Aerospace Controls Laboratory
Massachusetts Institute of Technology
33-326, 77 Massachusetts Ave.
Cambridge, MA 02139

April 15, 2008

Re: Final Technical Report for FA9550-04-1-0458

To Whom it May Concern,

The final report for this project consists of the attached documents, which include the three reports submitted each year of the program, and three recent Journal papers on research supported by the project (two published, 1 just submitted).

Yours truly,

Jonathan How

Jonathan P. How
Professor

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
15-04-2008

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
01 SEP 2004 - 29 FEB 2008

4. TITLE AND SUBTITLE
CONTROL ARCHITECTURE DESIGN AND DEMONSTRATION FOR
COOPERATIVE UAV'S

5a. CONTRACT NUMBER
FA9550-04-1-0458

5b. GRANT NUMBER
AFOSR

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. Jonathan P. How

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology
Department of Aeronautics and Astronautics
Cambridge, MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NL
875 Randolph
Arlington VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-SR-AR-TR-08-0385

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The research objective is to develop algorithms that can be embedded in a distributed coordination and control architecture for teams of multiple UAVs. The main problem is to determine the role of errors in the situational awareness (SA) in the fleet coordination problem by investigating two fundamental questions: how does specific sensor information impact the estimation error, and how does this estimation error impact the control performance? The answers to these questions will provide a clear indication of the impact of errors in the SA on the control performance and should enable the design of an efficient (and dynamic) communication architecture. Our research has resulted in several algorithms that use mixed-integer linear programming (MILP) to perform both the activity and path planning components of the team coordination. The focus of recent work has been on improving the robustness of the various planning algorithms to uncertainty in the information available. We have also added four new UAVs to the multi-UAV testbed that is being used to evaluate various distributed control architectures.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			
					19b. TELEPHONE NUMBER *(Include area code)*

Reset

CONTROL ARCHITECTURE DESIGN AND DEMONSTRATION FOR COOPERATIVE UAV'S

AFOSR # FA9550-04-1-0458

Jonathan P. How
Aerospace Controls Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

Abstract

The research objective is to develop algorithms that can be embedded in a distributed coordination and control architecture for teams of multiple UAVs. The main problem is to determine the role of errors in the situational awareness (SA) in the fleet coordination problem by investigating two fundamental questions: how does specific sensor information impact the estimation error, and how does this estimation error impact the control performance? The answers to these questions will provide a clear indication of the impact of errors in the SA on the control performance and should enable the design of an efficient (and dynamic) communication architecture. Our research has resulted in several algorithms that use mixed-integer linear programming (MILP) to perform both the activity and path planning components of the team coordination. The focus of recent work has been on improving the robustness of the various planning algorithms to uncertainty in the information available. We have also added four new UAVs to the multi-UAV testbed that is being used to evaluate various distributed control architectures.

Research Status

The following progress has been made in the past year:

1. Developed new distributed task assignment algorithm that removes inconsistencies due to imperfect information across the UAV team by communicating the plans to resolve conflicts [1]. Simulation results show that the new approach is more efficient than simply trying to fully synchronize the situational awareness.
2. Developed new theoretical approach to optimal search that captures the uncertainty typically present in a priori descriptions of the targets and threats in the environment [2].
3. Extended our Robust Model Predictive Controller (RMPC) [3] algorithm to develop a "Robust Safe but Knowledgeable" (RSBK) approach [4]. Compared our RMPC algorithm with a LMI-based min-max approach, and developed a hybrid algorithm that retains the advantages of both [5].
4. Augmented the previous UAV fleet with 4 Monocoupe 90As [9].
5. Performed an experimental demonstration of our decentralized MPC algorithm using three rovers [6,7]. Extended the RMPC algorithm to a variable horizon formulation that is less restrictive for vehicle maneuvering problems [8].

The following sections provide more detail on the main topics.

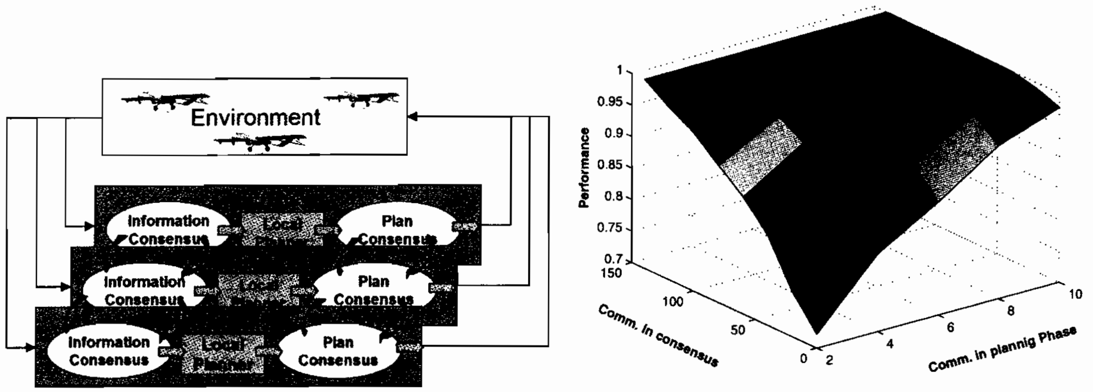


Fig. 1: *Left: Robust Distributed Task Assignment algorithm showing the two stages of communication used. Right: Comparison of effectiveness of communication in the 2 phases of the RDTA algorithm (consensus and planning) on the performance.*

Distributed Task Assignment: Recent work has focused on developing a new *robust distributed task assignment* (RDTA) algorithm for a team of UAVs. Centralized planning can be impractical due to constraints on the computation (algorithm scalability) and communication (robustness). However, task planning is hard to decentralize because of the strong coupling between agents in the choice of the tasks. Distributed task allocation methods have been developed that synchronize information across the fleet, have each agent plan by running a centralized method, and then implement their own resulting plan, but this "implicit coordination" approach can easily fail by giving conflicting plans if the synchronization is poor, so it places stringent requirements on the communication resources [1]. Furthermore, reaching consensus is not always possible for large teams in a complex, rapidly changing environment. We have extended the basic implicit coordination approach to achieve better (i.e., more consistent) performance with imperfect data synchronization. The resulting Robust Distributed Task Assignment method shown in Fig. 1 assumes some degree of data synchronization, but adds a second planning step based on shared planning data. This exchange of candidate plans guarantees that all UAVs have the same information during the final planning step, which ensures that the independently designed plans are consistent. The approach is analogous to closing a synchronization loop on the planning process to reduce the sensitivity to exogenous disturbances.

The performance of the algorithm was investigated for a scenario with 5 UAVs and 10 targets to show the advantages of this method in reducing the conflicts in the assignments, generating feasible plans and increasing the performance of the plan compared to implicit coordination. Fig. 1 compares the effect of communication (measured in *words*) in the different phases of the planning (reaching consensus in SA, communicating plans) on the performance of the assignment. Note that in this plot, implicit coordination is effectively equivalent to communicating 2 words, or one petal, in the planning phase. The graph shows that increasing the communication during either the consensus or planning phases improves the team performance. To maximize the performance, it is clear that some communication in both phases is needed, but the results also clearly show that communication in the planning phase is more efficient than in the information phase in the sense that 8 words of communication in the planning phase have approximately the same effect on performance as 80 words in the consensus phase. This improved efficiency indicates that RDTA provides a tractable distributed planning approach for complex task allocation problems.

Robust Search: We have extended traditional search algorithms to account for uncertainty in the initial information. Most, if not all, previous research assumes that the initial target information (such as the probability of target existence in each grid cell) is known precisely. This assumption is very restrictive, particularly in highly uncertain environments, because poor/conflicting intelligence, communication delays, or noisy sensors will cause uncertainty in the information. Our algorithm relaxes the assumption on the precision of the initial knowledge by generalizing the prior information to be a *distribution on the initial distribution* [2]. The Beta distribution ($f(x) = \frac{\Gamma(b+c)}{\Gamma(b)\,\Gamma(c)}\, x^{b-1}(1-x)^{c-1}$, $x \in [0,1]$) is used to describe the uncertainty in the initial probability (i.e., the support is the prior probability of target existence). A key benefit of using Beta distributions is the analytical tractability since, by assuming a very general Bernoulli sensor model, the Beta distribution is a *conjugate* distribution, and therefore the Beta distribution property is preserved through a Bayesian measurement update.

This new search formulation was used [2] to develop analytic predictions of the mean time to look at a target in order to raise the confidence of a target existence beyond a predefined threshold α. This metric (also called, looks or glints) is useful for planning because, given the prior uncertainty description of the prior probability, an estimate for the UAV "time on target" can be obtained. For example, consider a scenario with 4 targets in the environment and an assumed initial probability of existence of 50%; a target is considered detected if the probability exceeded a confidence of $\alpha = 0.85$; and the sensor error was assumed to be $\xi = 0.95$ (probability of correct detection). Using the mean time on target for a nominal planning scheme (column 2 in the table), each target exceeds the prescribed threshold for a 280-step mission. Our new robust algorithm developed a 196-step mission that exceeded the confidence for each target, a reduction of 30% from the nominal.

Mean time on target: $\alpha = 0.85, \xi = 0.95$

Target	α_{Nom}	α_{New}
1	0.89	0.85
2	0.90	0.85
3	0.87	0.85
4	0.85	0.85
Total Time	280	196

Trajectory Optimization: Model Predictive Control (MPC) is a powerful tool for the UAV guidance problem because optimization-based controllers can operate close to constraint boundaries to obtain better performance than traditional control approaches. As a result, however, small disturbances could drive the system infeasible, so we must account for external disturbances and modeling errors. We have developed a new robust MPC (RMPC) approach that uses *constraint tightening* (CT) [15] with a more general candidate policy, thereby leading to a less constrained optimization and hence a less conservative controller [3,7,8,10]. The approach retains "margin"

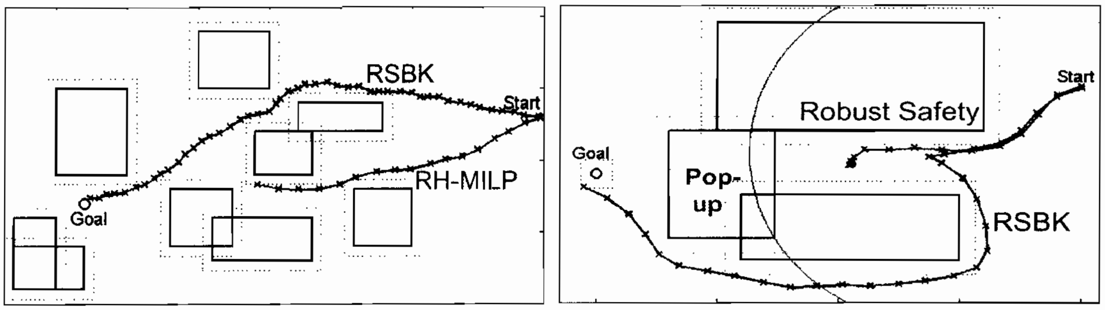


Fig. 2: *RSBK compared with previous algorithms, which fail to navigate the UAV to the goal due to disturbances or local minimum.*

for future feedback action, which becomes available to the MPC optimization as time progresses. Since robustness follows only from the constraint modifications, only nominal predictions are required, avoiding both the large growth in problem size associated with incorporating multivariable uncertainty in the prediction model and the conservatism associated with worst case cost predictions, a common alternative.

Robust Safe but Knowledgeable (RSBK) Algorithm: We extended the RMPC algorithm to design trajectories for a UAV performing long-range maneuvers [4]. RSBK uses a robust control invariant admissible set as the terminal set of each plan that does not need to be a target set of the overall guidance problem. The approach is similar to the previous RH-MILP algorithm, where the controller designs a detailed trajectory over a short planning horizon using a shortest-path approximation of the cost-to-go. For RSBK this cost-to-go estimate is based on the tightened constraints, so that the future maneuver is feasible, even with the disturbance inputs. Fig. 2 shows numerical simulations that highlight the advantages of the RSBK algorithm: the vehicle is guaranteed to remain safe under the influence of disturbances; much longer robust trajectories can be constructed on-line. Combined with our Decentralized MPC (DMPC) work [7], our new algorithm can be used to generate trajectories for multiple vehicles maneuvering in a complex and uncertain environment.

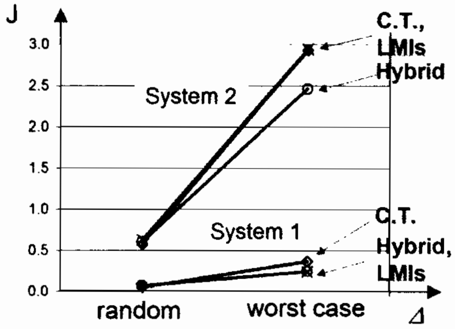


Fig. 3: *Perf comparison of 3 RMPC algorithms under different disturbance inputs*

Hybrid Robust MPC Algorithm: Another challenge associated with RMPC is to achieve better performance under worse-case disturbance inputs. We unified our constraint tightening RMPC algorithm with a min-max RMPC algorithm that uses Linear Matrix Inequalities (LMIs) and robust optimization techniques [16] to achieve robustness and performance [5]. Although apparently different in structure, we show that these two MPC algorithms are closely related, and actually have equivalent decision variables when performing a closed-loop prediction. Based on this analysis we developed a *hybrid algorithm* that retains the advantages of both: a) large feasible region indicating the algorithm can tolerate higher disturbance levels; b) performance guarantees under worst-case dis-

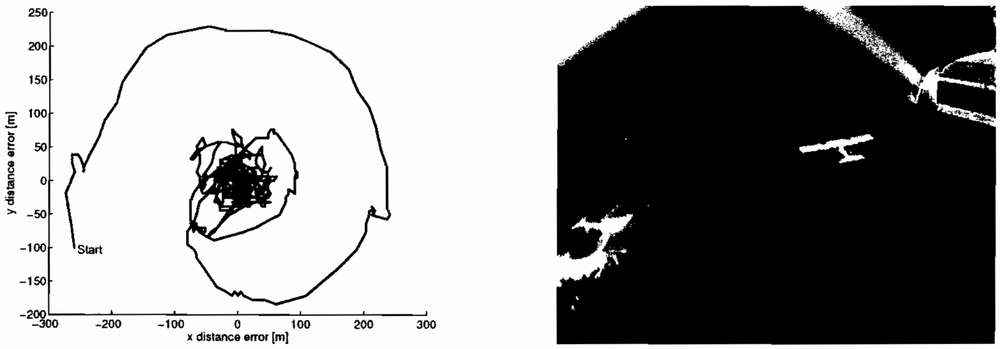


Fig. 4: *Flight data showing show the relative separation error between 2 UAVs during an autonomous in-flight rendezvous with a photo from onboard.*

turbance inputs. As shown in Fig. 3, the three algorithms have similar performance with random (but bounded) disturbance inputs. When the worst case disturbances are applied, however, the hybrid algorithm outperforms the CT algorithm because it uses a min-max cost. With a more constrained system (System 2), the hybrid algorithm achieves better performance than the LMI-based algorithm because its larger feasible set gives a larger operating region.

Multi-UAV Testbed: The UAV testbed has been operated autonomously on numerous occasions [9]. For example, Fig. 4 shows the relative position error for two of the UAVs performing an autonomous rendezvous. The position error is shown to converge to within 25 m. Wind disturbances during this flight were approximately 1 m/s (≈5% of flight speed). The relative position errors show the vehicles maintaining coordinated flight despite the moderate disturbance levels acting on the system. A second type of UAV has been added to the testbed using a Monocoupe airframe (see Fig. 5), which have several advantages over the Trainer 60 (double wing area allows for a payload capacity of up to 5 lbs and larger fuel capacity increases the flight time to 2 hours). These vehicles can also be purchased as almost ready to fly, so they retain the operational simplicity originally envisaged for the UAV testbed. The current testbed has four Monocoupes that fly with the Trainer 60's in a heterogeneous team.

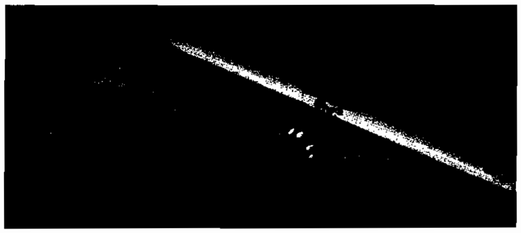


Fig. 5: *Comparison of Monocoupe and Trainer 60 in the MIT UAV testbed.*

Personnel Supported Professor Jonathan How; Graduate students Alighanbari, Kuwata (partial), and Harjes (partial); and Undergraduates: Engel, Abrahamsen, Burroughs, and Woodworth.

Publications See [1–14].

Honors/Awards Professor How was appointed as an Associate Fellow of the AIAA (Nov, 2004) and a Senior Member of IEEE (April, 2005).

AFRL Point of Contact: Lt. Col. Heise, AFOSR and Dr. Banda, AFRL/VACA

Transitions

- Jim Paduano at Nascent Technology (STTR April 2005).
- J. Wohletz at BAE/AlphaTech (STTR Oct 2004 – May 2005).
- Organized '05 ACC tutorial session on *Mixed-integer Programming for Control.*
- Dr. A. Richards (former student) is now a Lecturer in Controls and Dynamics, Dept. of Aerospace Engineering, University of Bristol

References

[1] M. Alighanbari and J. P. How, "Decentralized Task Assignment for Unmanned Aerial Vehicles," to appear at the IEEE *CDC*, Dec 2005.

[2] L. F. Bertuccelli and J. P. How, "Robust UAV Search for Environments with Imprecise Probability Maps," to appear at the IEEE *CDC*, Dec 2005.

[3] A. Richards and J. P. How, "Robust Model Predictive Control Using Generalized Constraint Tightening," submitted to the IEEE *Trans. on Auto. Control*, Feb 2005.

[4] Y. Kuwata, T. Schouwenaars, A. Richards, and J. P. How, "Robust Constrained Receding Horizon Control for Trajectory Planning," AIAA *Guidance, Navigation, and Control Conf.*, August 2005.

[5] Y. Kuwata and J. P. How, "Robust Constrained Model Predictive Control with Closed-loop Prediction: Constraints Tightening and LMIs," in preparation for the 2006 *ACC.*

[6] A. Richards and J. P. How, "Decentralized Model Predictive Control of Cooperating UAVs," presented at the IEEE *CDC*, 2004, pp. 4286–4291.

[7] A. Richards and J. P. How, "Robust Decentralized Model Predictive Control," submitted to *Automatica,* April 2005.

[8] A. Richards and J. How, "Robust Variable Horizon Model Predictive Control for Vehicle Maneuvering," submitted to *Int. J. of Robust and Nonlinear Control*, July 2005.

[9] E. King, Y. Kuwata, and J. P. How, "Experimental Demonstration of Coordinated Control for Multi-vehicle Teams," to appear, *Int. J. of Systems Science* 2005.

[10] A. Richards and J. P. How, "Robust Model Predictive Control with Imperfect Information" presented at IEEE *American Control Conference*, 2005, pp. 268-274.

[11] L. Bertuccelli, M. Alighanbari, J. P. How, "Robust planning for coupled, cooperative UAV missions," presented at the IEEE *CDC*, 2004, pp. 2917–2923.

[12] M. Alighanbari and J. P. How, "Cooperative Task Assignment of Unmanned Aerial Vehicles in Adversarial Environments" IEEE *ACC,* 2005, pp. 4661-4667.

[13] J. Deyst, S. Park, and J. P. How, "Lyapunov Stability of a Nonlinear Guidance Law for UAVs" to appear at the 2005 AIAA *Guidance, Navigation and Control Conf.*.

[14] A. Richards and J. P. How, "Implementation of Robust Decentralized Model Predictive Control," 2005 AIAA *Guidance, Navigation and Control Conf.*. AIAA-2005-6366

[15] L. Chisci, J. Rossiter and G. Zappa, "Systems with Persistent Disturbances: Predictive Control with Restricted Constraints," *Automatica,* Vol. 37(7) 2001, pp. 1019-1028.

[16] J. Lofberg, *Minimax approaches to robust model predictive control,* Ph.D. dissertation, Linkoping University, 2003.

New discoveries, inventions, or patent disclosures None.

Acknowledgment/Disclaimer Work was sponsored (in part) by the Air Force Office of Scientific Research, USAF, under grant FA9550-04-1-0458. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

CONTROL ARCHITECTURE DESIGN AND DEMONSTRATION FOR COOPERATIVE UAV'S

AFOSR # FA9550-04-1-0458

Jonathan P. How
Aerospace Controls Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

Abstract

The research objective is to develop algorithms that can be embedded in a distributed coordination and control architecture for teams of multiple UAVs. The main problem is to determine the role of errors in the situational awareness (SA) in the fleet coordination problem by investigating two fundamental questions: how does specific sensor information impact the estimation error, and how does this estimation error impact the control performance? The answers to these questions will provide a clear indication of the impact of errors in the SA on the control performance and should enable the design of an efficient (and dynamic) communication architecture. Our research has resulted in several algorithms that use mixed-integer linear programming (MILP) to perform both the activity and path planning components of the team coordination, and recent work has focused on improving the robustness of these algorithms to uncertainty in the SA. We have also implemented these algorithms on our external multi-UAV testbed and a new indoor multi-UAV testbed.

Research Status

The following progress has been made in the past year:

1. Demonstrated the benefits of our distributed task assignment algorithm for sparse communication networks. Extended previous consensus algorithms to remove the bias in the estimates for general network architectures [12,16,19,5].
2. Developed new theoretical approach to optimal search that captures the uncertainty typically present in a priori descriptions of the targets and threats in the environment [13,15,21,24].
3. Developed new computationally efficient decentralized MPC (DMPC) algorithms for multi-vehicle systems coupled through their constraints, which is the case for collision avoidance and formation flying scenarios. These techniques maintain robust feasibility of the entire team using local communication and computation, and thus scale well with the size of the UAV team [1,17,4,18,3,7].
4. Extended the DMPC algorithms to enable cooperation between vehicles without substantially increasing the sub-problem size [14]. Results indicate that this extension has the potential of significant performance improvements over previous techniques [22].
5. Performed an experimental demonstration of our distributed MPC algorithm through its distributed implementation on three rovers [7].

The following sections provide more detail on the main topics.

Distributed Planning: Previous reports presented a new *robust distributed task assignment* (RDTA) algorithm for a team of UAVs. We extended the analysis of that algorithm to consider the performance with different communication architectures and errors. The results show that the second communication step introduced during the planning phase of RDTA is crucial for very sparse networks because the convergence rate of the information consensus algorithms tends to be quite slow [12,19].

Developing a consensus for a team of agents with inconsistent information is a core component of many decentralized planning schemes, including RDTA. Numerous researchers have investigated this problem, and recent results proposed a Kalman consensus algorithm (KCA) and gave a detailed analysis of its convergence. However, we demonstrated that this KCA can result in biased estimates that deviate from the centralized solution, if it had been computed. The bias is shown to be related to a "double counting" of the information from agents that have a higher outflow (connectivity to the other agents) in the network. We modified the algorithm to correctly handle the differences in outflows in general networks, and the proof of convergence for this *Modified Distributed Kalman Consensus* (MDKC) algorithm to the unbiased estimate is provided for both static and dynamic communication networks [16,5].

Robust Planning: We developed a new robust approach to the task assignment of UAVs operating in uncertain dynamic environments for which the optimization data, such as target cost and target-UAV distances, are time-varying and uncertain [6,23]. The impact of this uncertainty in the data is mitigated by combining robust and adaptive planning, resulting in the *Robust Filter-Embedded Task Assignment* (RFETA) algorithm that uses both proactive techniques that hedge against the uncertainty and reactive approaches that limit churning behavior by the vehicles. Typical simulation results for a small problem are shown in Figure 1, which plots the actual accumulated score during a mission. Compared to the nominal algorithm, the robust strategy achieves a higher final accumulated value, which is consistent with selecting better task assignments given the uncertain data. The original FETA algorithm gives a faster convergence to the final value, which is consistent with avoiding churning, but the RFETA approach gives the same fast convergence, a higher final score, and the least standard deviation, and thus is the best overall strategy.

Robust Search: The goal of this research is robust assignment of search tasks in the presence of dynamic and uncertain target motion [13,15,21,24]. The approach taken is probabilistic, based on discrete-state, discrete-time Markov chain-like models. The target motion across the discretized environment is described by a probabilistic state transition matrix, where the transition to adjacent cells is fully represented by probability distributions. While the probabilities used to encode information about the environment are typically assumed to be exactly known in the search theory literature, they are often the result of prior information that is both erroneous and delayed, and will likely be poorly known to mission designers. Our work has developed a new framework that accounts for this uncertainty in the probability maps for stationary targets, and we recently extended the approach to consider dynamic environments.

The dynamic case considers uncertainty in the prior information, and creates *Un-*

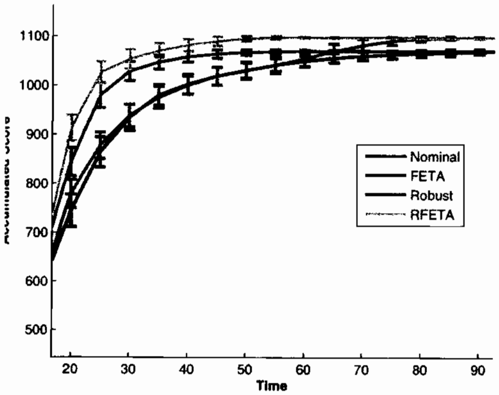


Fig. 1: Accumulated score and associated confidence level for four task planning algorithms.

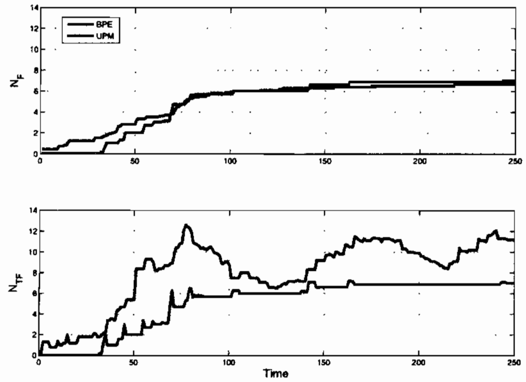


Fig. 2: Number of targets found (top) and thought to be found (bottom). BPE in blue, UPM in red.

certain Probability Maps (UPMs) that take into account both poor knowledge of the probabilities and the propagation of their uncertainty through the environment. Many methods could be chosen to describe the uncertainty, such as a truncated Gaussian distribution, but these distributions must be chosen wisely because the measurement update following the prediction can be computationally intensive if the distributions are not conjugate and sampling methods are needed to generate the posterior distribution. Our choice of the Beta distribution [13] to describe the uncertainty is conjugate with the Bernoulli sensor model, and thus computationally efficient. This new approach still requires an approximation in the prediction step that is similar to other techniques, but this approximation does not add any significant computational load because it can be written as a closed-form update to the mean and variance of the Beta distributions associated with each cell. As a result, the prediction and measurement update stages can be expressed as a recursion on the means and variances of the distributions, which are very simple to implement and conceptually very similar to conventional filtering methods.

A key result of this work is a new algorithm for implementing UPMs in real-time, and it was demonstrated in various simulations that this algorithm leads to more cautious information updates than previous approaches, and thus is less susceptible to false alarms. For example, Figure 2 gives results for a simple scenario showing that our UPM formulation creates a more realistic search strategy than the traditional Bayes update scheme (BPE), which is overly optimistic [15]. These results also provide numerous insights on the effect of the design parameters on the responsiveness of the new algorithm. This work has been extended to embed the uncertainty in the probabilistic target motion using Dirichlet distributions, and shows significant improvements in the search performance over otherwise mismatched motion models [24].

Trajectory Optimization: We extended the *robust safe but knowledgeable* (RSBK) algorithm [7–9,11,17,18] to the multi-vehicle case. RSBK has the advantage that it enables the use of much shorter planning horizons while still preserving the robust feasibility guarantees of our previous MPC approaches. A distributed version was also developed (called DRSBK), which is more suitable for real-time execution, retains the robust feasibility guarantees of the centralized approach, and only requires that each agent have local knowledge of the environment and neighbor vehicles plans. DRSBK also facilitates the use of a significantly more general implementation architecture for the distributed trajectory optimization, which further decreases the delay due to computation time [7,17].

Extending that work further, we developed a new decentralized trajectory optimization approach for systems with independent dynamics but coupled constraints [14,22]. The goal was to improve the performance of the entire fleet by including more cooperation between the vehicles. This is achieved by exploiting the sparse structure of active couplings that is inherent in many of the trajectory optimization problems of interest. This enables each local optimization to use a low-order parameterization of the other agents states, thereby facilitating negotiation while keeping the problem size small. The key features of this approach include (a) no central negotiator is required; (b) it maintains feasibility over the iterations, so the algorithm can be stopped at any time; and (c) the local optimizations are shown to always decrease the overall cost.

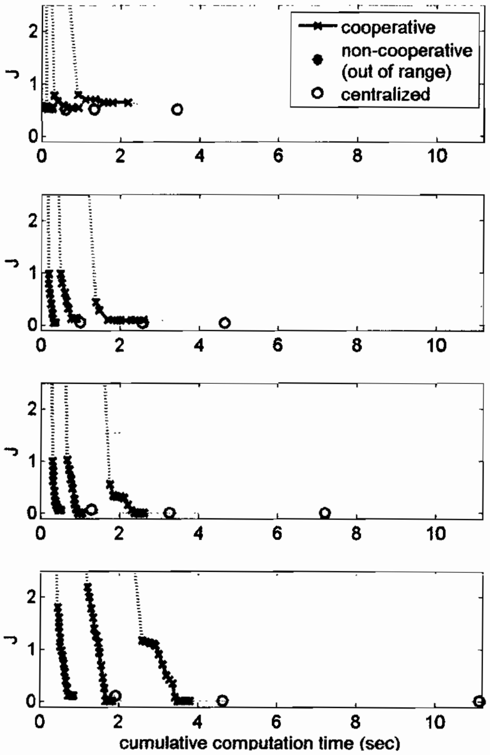


Fig. 3: Performance and computation comparison with $n_V = \{5, 7, 10, 15\}$ (top to bottom).

Figure 3 shows simulation results comparing the global objective value and cumulative computation time of three algorithms: centralized, (non-cooperative) decentralized, and the new cooperative decentralized. Different planning horizons $N = \{4, 6, 8\}$ and fleet sizes n_V were considered to investigate the scalability of the algorithms. The solutions of the decentralized non-cooperative approach have very high cost and, although the computation times are small, are off the plot. The lines with × show the evolution of the global cost of the cooperative decentralized algorithm. The results show that this new algorithm has objective values comparable to the centralized (and hence globally optimal) solutions (o), but the computation scales much better with the fleet size.

Multi-vehicle Testbeds: The multi-rover testbed was used to test the DRSBK algorithm in real-time. Numerous scenarios (e.g. Fig. 4) were constructed to highlight key features of the algorithm: on-board laptops generate trajectories on-line, which shows the computational advantages for real-time applications; the rovers are moving initially, which highlights the ability to quickly initialize the algorithms; the vehicles are required to maneuver in a constrained environment, which demonstrates the robust feasibility under the action of disturbances and that plans based on dis-

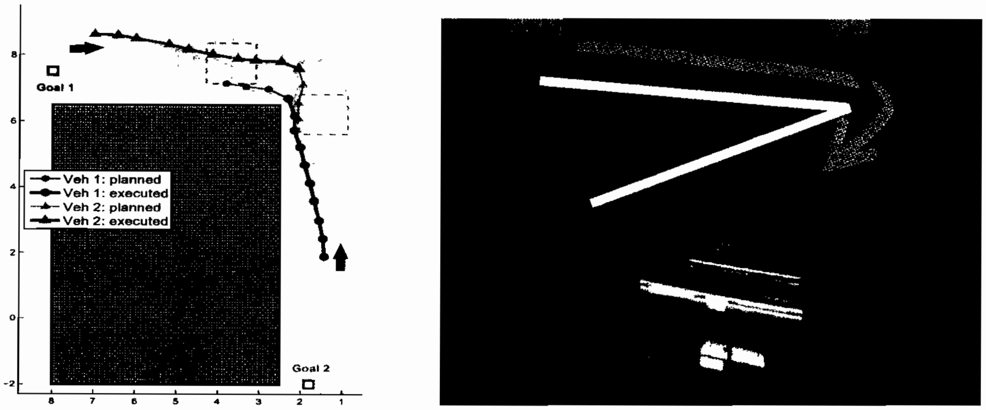


Fig. 4: DRSBK demonstration for obstacle and vehicle avoidance

tributed computation can satisfy the coupling collision avoidance constraints. The results also demonstrated online dynamic grouping and re-grouping of the vehicles, enabling parallel computation.

Fig. 5: Coupe UAVs, Coupe's flying, Coupe pan/tilt camera

The UAV testbed (Figs. 5 and 6) is currently being used to perform research on autonomous search and target tracking [2]. The hardware is being upgraded to carry additional onboard computers to increase the level of autonomy, provide onboard vision processing, and support ad-hoc communication networks between vehicles. The coordination algorithms described in the report are also being demonstrated on a unique indoor testbed that includes multiple flying and ground vehicles [20].

Fig. 6: Automatic video tracking antenna.

Acknowledgment/Disclaimer: Work was sponsored (in part) by the Air Force Office of Scientific Research, USAF, under grant FA9550-04-1-0458. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

References

[1] A. Richards and J. How, “Robust Variable Horizon Model Predictive Control for Vehicle Maneuvering,” *Int. J. of Robust and Nonlinear Control*, Vol. 16, March 2006, pp. 333-351.

[2] E. King, Y. Kuwata, and J. How, "Experimental Demonstration of Coordinated Control for Multi-vehicle Teams," *Int. J. of Systems Science*, Vol. 37, No. 6, May 2006, pp. 385–398.

[3] A. Richards, L. Breger, and J. How, "Analytical Performance Prediction for Robust Constrained Model Predictive Control," *Int. J. of Control*, Vol. 79, No. 8, Aug. 2006, pp. 877-894.

[4] A. Richards and J. How, "Robust Model Predictive Control Using Generalized Constraint Tightening," in revision for IEEE *Trans. on Automatic Control*, Nov. 2005.

[5] M. Alighanbari and J. How, "An Unbiased Kalman Consensus Algorithm," submitted to IEEE *Trans. on Automatic Control*, Feb. 2006.

[6] L. Bertuccelli, M. Alighanbari, J. How, "A Robust Approach to the UAV Task Assignment Problem," submitted to *Int. J. of Robust and Nonlinear Control*, May 2006.

[7] Y. Kuwata, A. Richards, T. Schouwenaars, and J. How, "Distributed Robust Receding Horizon Control for Multi-vehicle Guidance," submitted to IEEE *Trans. on Control System Tech.*, 2006.

[8] A. Richards and J. P. How, "Robust Decentralized Model Predictive Control," submitted to the *International Journal of Control* July 2006.

[9] A. Richards and J. How, "Implementation of Robust Decentralized Model Predictive Control," *GNC* AIAA-2005-6366

[10] J. Deyst, J. How, and S. Park, "Lyapunov Stability of a Nonlinear Guidance Law for UAVs" *Atmospheric Flight Mechanics Conference and Exhibit.* AIAA-2005-6230.

[11] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How, "Robust Constrained Receding Horizon Control for Trajectory Planning," *GNC* AIAA-2005-6079.

[12] M. Alighanbari and J. How, "Decentralized Task Assignment for Unmanned Aerial Vehicles," IEEE *CDC* , 2005, pp. 5668–5673.

[13] L. Bertuccelli and J. How, "Robust UAV Search for Environments with Imprecise Probability Maps," IEEE *CDC* , 2005, pp. 5680–5685.

[14] Y. Kuwata and J.P.How, "Decentralized Cooperative Optimization for Systems Coupled through the Constraints," *Conf. on Cooperative Control and Optimization,* Jan, 2006.

[15] L. Bertuccelli and J. How, "Search for Dynamic Targets with Uncertain Probability Maps" IEEE *ACC*, 2006.

[16] M. Alighanbari and J. How, "An Unbiased Kalman Consensus Algorithm" *ACC*, 2006.

[17] Y. Kuwata, A. Richards, T. Schouwenaars, and J. How, "Decentralized Robust Receding Horizon Control for Multi-vehicle Guidance" IEEE *ACC*, 2006.

[18] A. Richards and J. How, "Robust Stable Model Predictive Control with Constraint Tightening" IEEE *ACC*, 2006.

[19] M. Alighanbari and J.How, "Robust Decentralized Task Assignment for Cooperative UAVs," to appear at *GNC* AIAA-2006-6454.

[20] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron, "Indoor Multi-Vehicle Flight Testbed for Fault Detection, Isolation, and Recovery," to appear at AIAA *GNC,* 2006.

[21] L. Bertuccelli and J.How, "Bayesian Forecasting in Multi-vehicle Search Operations," to appear at *GNC,* AIAA-2006-6460

[22] Y. Kuwata and J. How, "Decentralized Cooperative Trajectory Optimization for UAVs with Coupling Constraints," to appear at IEEE *CDC*, 2006.

[23] M. Alighanbari, L. F. Bertuccelli and J. How, "A Robust Approach to the UAV Task Assignment Problem," to appear at IEEE *CDC*, 2006.

[24] L. F. Bertuccelli and J. How, "UAV Search for Dynamic Targets with Uncertain Motion Models," to appear at IEEE *CDC*, 2006.

Personnel Supported: Professor Jonathan How; Graduate students Alighanbari, Kuwata, and Bertuccelli (partial); and Undergraduates: Engel and Woodworth.

Publications: See [1]–[24].

Honors/Awards: Professor How was appointed as an Associate Fellow of the AIAA (Nov, 2004) and a Senior Member of IEEE (April, 2005).

AFRL Point of Contact: Lt. Col. Heise, AFOSR and Dr. Banda, AFRL/VA

Transitions:

- RDTA transferred to J. Paduano at Nascent Technology (STTR 2005).
- RDTA transferred to T. Vaneck at Aurora Flight Sciences Corp. (SBIR 2006).
- MILP and DRSBK transferred to R. Miller at Northrop Grumman (May 2006).

New discoveries, inventions, or patent disclosures: None.

CONTROL ARCHITECTURE DESIGN AND DEMONSTRATION FOR COOPERATIVE UAV'S

AFOSR # FA9550-04-1-0458

Jonathan P. How
Aerospace Controls Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

Abstract

The research objective is to develop algorithms that can be embedded in a distributed coordination and control architecture for teams of multiple UAVs. The goal is to determine the role of errors in the situational awareness (SA) in the fleet coordination problem by investigating two fundamental questions: how does specific sensor information impact the estimation error, and how does this estimation error impact the control performance? The answers to these questions should provide a clear indication of the impact of errors in the SA on the control performance and should enable the design of an efficient (and dynamic) communication architecture. Our research has resulted in new algorithms to perform distributed task assignment that is robust to poor network connectivity, cooperative decentralized path planning that is robust to environmental disturbances, and multi-UAV search that is robust to uncertainty in the knowledge of the environment. We have implemented many of these algorithms on our new indoor multi-UAV testbed called RAVEN.

Research Status

The following progress has been made in the past year:

1. Further analysis of our Unbiased Kalman-based Consensus Algorithm [4, 9, 10, 11]. Developed a new, completely decentralized, auction-based task assignment algorithm that creates conflict free assignments with limited communication and without relaying of information between agents. The algorithm is proven to converge and is shown to be resource efficient [12].
2. Extended prior work on robust MPC [1, 3, 6, 7, 16, 17] using constraint tightening by generalizing the feedback correction policy. The new approach is shown to represent a strictly larger class of feedback policies when compared to previous algorithms [19]. We also developed a new off-line convex optimization procedure to design a disturbance rejection controller that can tolerate much stronger disturbances leading to significant performance improvements over previous approaches at high disturbance levels [19, 21].
3. Developed a new cooperative decentralized optimization algorithm [6, 17] that minimizes the global performance by solving a series of small subproblems across the team without just reproducing the global optimization problem on each vehicle. The global objective is proven to monotonically decrease over each iteration, and the feasibility of the entire team is maintained. The algorithm was extended

to develop a cooperative form of the decentralized robust safe but knowledgeable (DRSBK) algorithm for multi-vehicle trajectory optimization [18]. CDRSBK achieves cooperative behaviors by enabling vehicles to sacrifice the local cost if it leads to a global improvement [18, 20, 21].

4. Experimental demonstration of our distributed planning algorithms [21, 22] on a new indoor UAV testbed [24, 26].

The following sections provide more detail on the main topics.

Distributed Planning: Developing consensus for a team of agents with inconsistent information is a core component of many decentralized planning schemes. We recently extended the analysis of the Unbiased Kalman Consensus Algorithm (UKCA) [9, 12], with the following key observations: (i) exchanging both the state x_i and uncertainty weights P_i in each iteration enables UKCA to converge to the desired weighted average for a very general class of communication networks; (ii) the scaling factor introduced in UKCA is associated with the outflow of the sending agent, which differs from other approaches in the literature; (iii) this scaling also has the implicit effect of essentially setting the outflows of all agents to 1, but has no effect on the inflow. Thus the scaling does not just implicitly balance the network, and standard convergence results do not apply. We therefore provide a proof of the convergence of UKCA to the unbiased estimate for both static and dynamic communication networks [4, 9].

Many distributed task assignment algorithms (including RDTA [10, 12]) require that agents relay information to neighbors. Our new auction based task assignment (ABTA) algorithm combines consensus and auction ideas to create conflict free assignments using limited communication without requiring relaying of information. The basic idea of the ABTA is for each agent to act in a greedy way and choose the best task for itself. It then communicates its value for the tasks with its neighbors to determine if it is the best agent to be assigned to that specific task. If an agent then determines that another agent can achieve a better value by being assigned to a task, it discards the task and chooses the next best task. This is similar to other auction algorithms, but in ABTA the information is only exchanged between neighbors. ABTA is proven to converge to a conflict free complete assignment for very general communication networks. A comparison to implicit coordination shows that the performance is within 7% of the best assignment, but there is a significant reduction in the communication requirements [12].

Robust Search: We developed computationally tractable search algorithms that account for the uncertainty in the optimization parameters [13, 14, 15]. The work is based on Markov Chain theory and its applications to finite state, finite action, finite horizon Markov Decision Processes (MDPs). It gives a computationally tractable solution to finding the optimal policies subject to uncertainty in the initial probabilities and the state transition matrix. The novelty in this work is the adaptation of the Sigma Point sampling methods (SPSM) to dynamic multi-stage problems where the probabilistic parametrization may be poor. We show that the SPSM can be used to generate computationally economical samples of the uncertain transition matrices.

Furthermore, the probability distributions on the uncertain parameters can be maintained in a closed form approximation with virtually no additional computational overhead by compactly describing them with the Dirichlet distribution.

Distributed Trajectory Optimization: We developed a new form of robust Model Predictive Control (MPC) using constraint tightening, where the degree of tightening is a convex function of the feedback parameters [19]. The proposed approach is shown to be able to represent a strictly larger class of feedback policies when compared to previous algorithms. Further analytical results provide a) necessary and sufficient conditions on the choice of feedback parameters for the existence of a nonempty output constraint set; and b) a sufficient condition for the existence of a nonempty robust invariant set. Combined with the convex parametrization, this enables an off-line linear optimization to determine the feedback policy that can tolerate much stronger disturbances while robustly satisfying the constraints.

Simulation results show that this proposed controller leads to both a larger feasible set and performance improvements under the action of strong disturbances. For example, Fig. 1 shows a set of initial states from which a controller has a feasible solution, for three controllers `2-step nil`, `3-step nil`, and `max-dist`. In the top plot with $\beta = 1.5$, the controllers `3-step nil` and `max-dist` have a larger initial feasible state set compared to `2-step nil`. When the disturbance level $\beta = 1.89$, the controller `max-dist` has a feasible region that is much larger than controller `3-step nil`, as shown in Fig. 1 (bottom). The larger set allows for the system to operate in a larger region, directly affecting the performance of the controller [19].

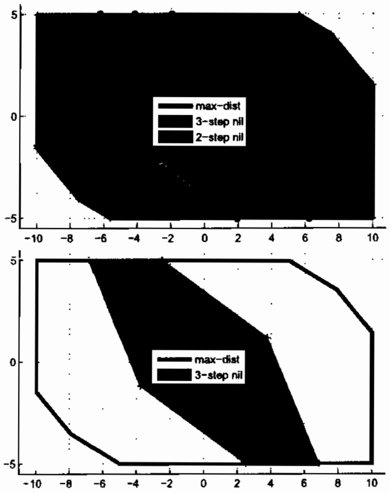


Fig. 1: Set of states for which a feasible solution exists.

We also extended prior work to develop a cooperative form of the distributed robust Model Predictive Control that is used for multi-vehicle trajectory optimization (called CDRSBK) [20]. The overall goal is to develop an approach that solves small subproblems but minimizes a fleet-level objective. In this new algorithm, vehicles solve their subproblems in sequence, while simultaneously generating feasible perturbations to the decisions of the other vehicles. In order to avoid reproducing the global optimization, the decisions of other vehicles are parameterized using a much smaller number of variables than in the centralized formulation. The resulting algorithm is shown to be robustly feasible under the action of unknown but bounded persistent disturbances and monotonically decreases the fleet objective while cycling through the vehicles in the fleet and over the time.

To demonstrate the algorithm, consider a scenario with two vehicles trying to reach their own targets while avoiding obstacles and each other. Fig. 2 shows the trajectories generated by the original DRSBK and cooperative CDRSBK algorithms. Because UAV 1 has to traverse a longer route, the best solution is for UAV 2 to "move over".

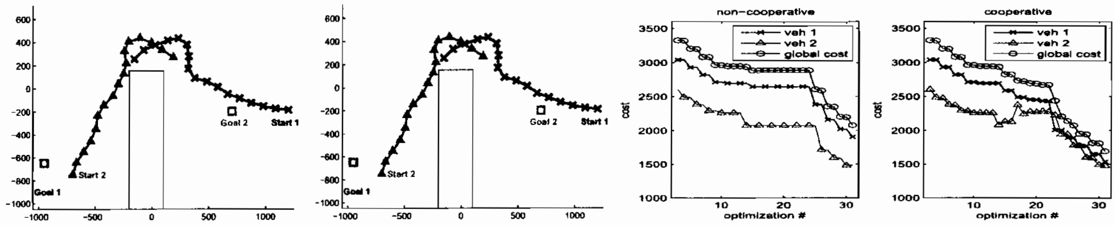


Fig. 2: Trajectories executed using DRSBK (left); CDRSBK (middle left); and time history of the objective function

While both distributed algorithms maintain feasibility under the action of disturbances, only the CDRSBK algorithm naturally recovers this cooperative behavior. Fig. 2 also shows the time history of the individual cost J^p and the fleet cost J. Both algorithms monotonically decrease the fleet objective, but as shown in the far right figure, the cooperative formulation allows the individual cost to increase if it leads to a larger improvement of the fleet cost. Between optimization 14–17, UAV 2 yields to the vehicle with a worse cost (UAV 1), enabling a large reduction in the fleet cost J. The average computation time for this scenario was 0.050s for DRSBK and 0.064s for CDRSBK. These simulation results clearly demonstrate the proposed algorithm can improve the fleet objective by temporarily sacrificing on the individual objective, with only a minimal impact on the computation time.

Multi-vehicle Testbeds: We have developed a unique indoor multi-vehicle test facility called RAVEN (Real-time indoor Autonomous Vehicle test ENvironment) to study long-duration missions in a controlled environment [24, 22]. A key feature of RAVEN is the Vicon global metrology system. By attaching reflective balls to the vehicle's structure, the Vicon MX Camera system and Tarsus software can track and compute the vehicle's position and attitude information at rates up to 120 Hz, with a 10 ms delay, and sub-mm accuracy. Just as GPS spurred the development of large-scale UAVs, we expect this new sensing capability to have a significant impact on 3D indoor flight, which has historically been restricted to very small volumes.

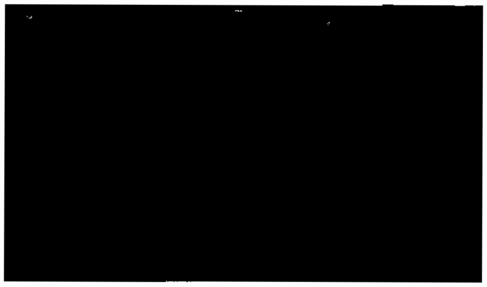


Fig. 3: 10 UAVs with 1 operator.

RAVEN is an excellent rapid prototyping environment for UAV research. In addition to demonstrating advanced path planning concepts (e.g., CDRSBK test flights) [21, 22] and flying large UAV teams (see Fig. 3), it has also been used for multi-UAV search and track using onboard vision [27, 28] (see Figs. 4–5). In this case, the high-level mission management system ensures that an adequate number of UAVs are available to search and track targets in unknown locations, while a cooperative algorithm was used to generate accurate estimates of the target locations by fusing sensor data from multiple UAVs (see Fig. 5) [27].

Fig. 6 shows an example of rapid prototyping done on aggressive flight manoeuvres using RAVEN [26]. The primary objective of this work is to design hybrid nonlinear

Fig. 4: Multi-UAV search and track

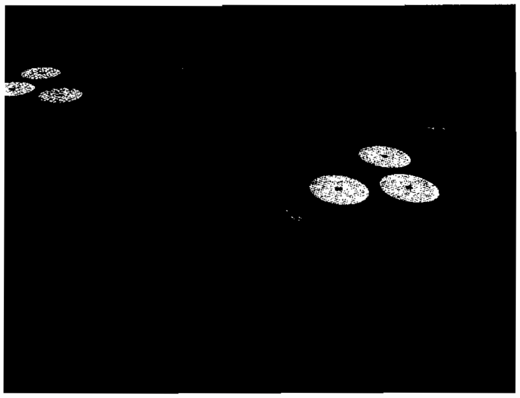


Fig. 5: Persistent tracking of UGV

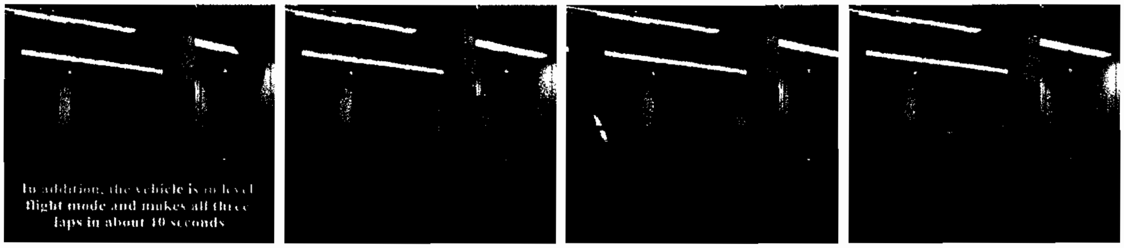

In addition, the vehicle is in level
flight mode and makes all three
laps in about 10 seconds

Fig. 6: Autonomous aircraft hover, transition to level flight and back to hover.

controllers to execute very agile aerobatics, similar to that expected for future MAV operations. Fig. 6 shows a sequence that illustrates the current capabilities (vertical take-off, hover, transition to horizontal flight, tracking a tight circular path, transition back to vertical) [25, 26]. Videos are available at `http://aerobatics.mit.edu`.

Acknowledgment/Disclaimer: Work was sponsored (in part) by the Air Force Office of Scientific Research, USAF, under grant FA9550-04-1-0458. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

References

[1] A. Richards and J. How, "Robust Variable Horizon Model Predictive Control for Vehicle Maneuvering," *Int. J. of Robust and Nonlinear Control*, Vol. 16, March 2006, pp. 333-351.

[2] E. King, Y. Kuwata, and J. How, "Experimental Demonstration of Coordinated Control for Multi-vehicle Teams," *Int. J. of Systems Science*, Vol. 37, No. 6, May 2006, pp. 385–398.

[3] A. Richards, L. Breger, and J. How, "Analytical Performance Prediction for Robust Constrained Model Predictive Control," *Int. J. of Control*, Vol. 79, No. 8, Aug. 2006, pp. 877-894.

[4] M. Alighanbari and J. How, "An Unbiased Kalman Consensus Algorithm," submitted to IEEE *Trans. on Automatic Control*, Feb. 2006.

[5] M. Alighanbari and J. P. How, "A Robust Approach to the UAV Task Assignment Problem," to appear *International Journal of Robust and Nonlinear Control*, July 2007.

[6] Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How, "Distributed Robust Receding Horizon Control for Multi-vehicle Guidance," IEEE *Transactions on Control System Technology*, Vol. 15, No. 4, July 2007, pp. 627–641.

[7] A. Richards and J. P. How, "Robust Decentralized Model Predictive Control," to appear *International Journal of Control* June 2007.

[8] S. Park, J. Deyst, and J. P. How, "Performance and Lyapunov Stability of a Nonlinear Path Following Guidance Method," to appear in the AIAA *JGCD*, April 2007.

[9] M. Alighanbari and J. How, "An Unbiased Kalman Consensus Algorithm" *ACC*, 2006, pp. 3519–3524.

[10] M. Alighanbari and J.How, "Robust Decentralized Task Assignment for Cooperative UAVs," AIAA *GNC* AIAA-2006-6454.

[11] M. Alighanbari, L. F. Bertuccelli and J. P. How, "A Robust Approach to the UAV Task Assignment Problem," IEEE *CDC*, Dec. 2006, pp. 5935-5940.

[12] M. Alighanbari, *Robust and Decentralized Task Assignment Algorithms for UAVs,* Ph.D. Thesis from the Department of Aeronautics and Astronautics, MIT, August 2007.

[13] L. Bertuccelli and J. How, "Search for Dynamic Targets with Uncertain Probability Maps" IEEE *ACC*, 2006.

[14] L. Bertuccelli and J.How, "Bayesian Forecasting in Multi-vehicle Search Operations," AIAA *GNC,* AIAA-2006-6460

[15] L. F. Bertuccelli and J. P. How, "UAV Search for Dynamic Targets with Uncertain Motion Models," presented at the IEEE *Conference on Decision and Control*, Dec. 2006, pp. 5941-5946.

[16] A. Richards and J. P. How, "Robust Stable Model Predictive Control with Constraint Tightening" IEEE *ACC*, June 2006, pp. 1557–1562.

[17] Y. Kuwata, A. Richards, T. Schouwenaars, and J. How, "Decentralized Robust Receding Horizon Control for Multi-vehicle Guidance" IEEE *ACC*, 2006.

[18] Y. Kuwata and J. P. How, "Decentralized Cooperative Trajectory Optimization for UAVs with Coupling Constraints," IEEE *CDC*, Dec. 2006, pp. 6820-6825.

[19] Y. Kuwata, A. Richards, and J. P. How, "Robust Receding Horizon Control using Generalized Constraint Tightening," IEEE *ACC*, July 2007, pp. 522–528.

[20] Y. Kuwata and J. P. How, "Robust Cooperative Decentralized Trajectory Optimization using Receding Horizon MILP," IEEE *ACC*, July 2007, pp. 4482–4288.

[21] Y. Kuwata, *Trajectory Planning for Unmanned Vehicles using Robust Receding Horizon Control,* Ph.D. Thesis from Department of Aeronautics and Astronautics, MIT, Dec. 2006.

[22] K. Culligan, M. Valenti, Y. Kuwata, and J. P. How, "Three-Dimensional Flight Experiments Using On-Line Mixed-Integer Linear Programming Trajectory Optimization," IEEE *ACC*, July 2007, pp. 5322-5328.

[23] E. Craparo, J. P. How, N. Roy and D. Pucci de Farias, "Decentralized Estimation under Communication Constraints," to appear at the AIAA *GNC*, August 2007.

[24] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron, "Indoor Multi-Vehicle Flight Testbed for Fault Detection, Isolation, and Recovery," AIAA *GNC,* 2006.

[25] J.P. How, "Multi-Vehicle Flight Experiments: Recent Results and Future Directions," AVT-146 Symposium on Platform Innovations and System Integration for Unmanned Air, Land and Sea Vehicles, Florence, Italy, May 2007.

[26] A. Frank, M. Valenti, D. Levine, and J. P. How, "Hover, Transition, and Level Flight Control Design for a Single-Propeller Indoor Airplane," to appear at the AIAA *GNC*, August 2007.

[27] B. Bethke, M. Valenti, J. P., How and J. Vian, "Cooperative Vision Based Estimation and Tracking Using Multiple UAVs", *Conference on Cooperative Control and Optimization,* Gainsville, FL, January 2007.

[28] Brett Bethke, *Persistent Vision-Based Search and Track Using Multiple UAVs* S.M. Thesis, Dept. of Aeronautics and Astronautics, MIT, June 2007.

Personnel Supported: Professor Jonathan How; Graduate students Kuwata, Alighanbari, and Bertuccelli (partial); and Undergraduates: Woodworth, Cohen.

Publications: See [1]–[28].

Honors/Awards: Professor How was appointed as an Associate Fellow of the AIAA (Nov, 2004), a Senior Member of IEEE (April, 2005), and promoted to Full Professor at MIT(July, 2007).

AFRL Points of Contact: LtCol Scott R. Wells, (AFOSR) and Dr. Banda (AFRL).

Transitions:

- RDTA, ABTA transferred to J. Paduano at Aurora (AFOSR STTR PhII 2007).
- RDTA transferred to T. Vaneck at Aurora (NASA SBIR PhI 2007).
- RDTA transferred to J. Vian at Boeing (Seattle), Nov. 2006

New discoveries, inventions, or patent disclosures: None.

Distributed Robust Receding Horizon Control for Multivehicle Guidance

Yoshiaki Kuwata, *Student Member, IEEE*, Arthur Richards, *Member, IEEE*, Tom Schouwenaars, and Jonathan P. How, *Senior Member, IEEE*

***Abstract*—This paper presents a new distributed robust model predictive control algorithm for multivehicle trajectory optimization and demonstrates the approach with numerical simulations and multivehicle experiments. The technique builds on the robust-safe-but-knowledgeable (RSBK) algorithm, which is developed in this paper for the multivehicle case. RSBK uses constraint tightening to achieve robustness to external disturbances, an invariant set to ensure safety in the presence of changes to the environment, and a cost-to-go function to generate an intelligent trajectory around known obstacles. The key advantage of this RSBK algorithm is that it enables the use of much shorter planning horizons while still preserving the robust feasibility guarantees of previously proposed approaches. The second contribution of this paper is a distributed version of the RSBK algorithm, which is more suitable for real-time execution. In the distributed RSBK (DRSBK) algorithm, each vehicle only optimizes for its own decisions by solving a subproblem of reduced size, which results in shorter computation times. Furthermore, the algorithm retains the robust feasibility guarantees of the centralized approach while requiring that each agent only have local knowledge of the environment and neighbhor vehicles' plans. This new approach also facilitates the use of a significantly more general implementation architecture for the distributed trajectory optimization, which further decreases the delay due to computation time.**

***Index Terms*—Cooperative control, distributed, invariant set, multivehicle experiments, model predictive control (MPC), robust feasibility.**

I. Introduction

Model predictive control (MPC) or receding horizon control (RHC) are natural techniques to approach to trajectory optimization problems for unmanned air vehicles (UAVs) because they can systematically handle constraints such as vehicle dynamics, flight envelope limitations, and no-fly zones [1]–[4]. MPC uses numerical optimization for online replanning, and a model of the system is embedded within the optimization to predict the future system behavior. A key advantage is that it can operate close to the constraint boundaries and obtain better performance than traditional approaches [5]–[7]. Recent research has focused on *robust MPC*, which is robust to external disturbances or inherent discrepancies between the model and the real process, and numerous techniques have been proposed in the past decade [8]–[15].

When using robust MPC in dynamic environments, fast online computation is needed in response to new information. However, the computation required scales poorly with both the length of the trajectories being planned and the number of vehicles to be planned for. This paper addresses both of these scalability issues, adopting shorter horizons for scaling with length and distributed computation for scalability with fleet size. The first contribution of this paper is the extension of the constraint tightening robust MPC [12], [16] to create a *robust-safe-but-knowledgeable* (RSBK) algorithm. This algorithm plans over only a short horizon, terminating in a robust control invariant set that needs not to be near the goal. A *cost-to-go* function is then used that provides a good estimate of the path beyond the planning horizon to the goal [2]. This combination enables a much faster computation than existing robustness methods [17], [18] that require the target be reachable within the planning horizon, which will require long horizons for long maneuvers.

For multivehicle control, decentralized MPC (DMPC) [19] addresses the computational issue associated with the centralized optimization by breaking the optimization into smaller subproblems, with the rationale that solving many small problems is faster and more scalable than solving one large problem. For multivehicle problems, it is natural to divide the problem such that the plan for each vehicle is computed on-board that vehicle, i.e., such that local decisions are made locally. Besides the computational advantages of DMPC, this also offers a reduction in the amount of data that needs to be exchanged between vehicles, and a potentially reduced level of dependency of any individual vehicle. The challenge of decentralized control is to ensure that distributed decision making leads to actions that are consistent with the actions of others and satisfy the coupled constraints. Various approaches have been investigated, including treating the influence of other subsystems as an unknown disturbance [20], coupling penalty functions [3], [4], [21], partial grouping of computations [22], loitering options for safety guarantees [23], and dynamic programming [24]. Some approaches involve iterative negotiations between subsystems [21], [25] and apply game theory to study convergence. Decentralization is further complicated when disturbances act on the subsystems, making the prediction of future behavior uncertain.

A second contribution of this paper is to develop a distributed form of RSBK (DRSBK). The primary computational benefit of the DRSBK algorithm over RSBK is that each vehicle only calculates its own trajectory, which is obtained by solving a subproblem of reduced size. The algorithm creates a queueing order of nonconflicting groups of vehicles, where each group optimizes sequentially, while vehicles within a group solve their subproblems in parallel. This does not require iteration, which is crucial

Manuscript received April 30, 2006; revised October 19, 2006. Manuscript received in final form April 2, 2007. Recommended by Guest Editor C.-Y. Su. This work was supported by by AFOSR under FA9550-04-1-0458 and DURIP F49620-02-1-0216.

Y. Kuwata, T. Schouwenaars, and J. P. How are with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: kuwata@mit.edu; tomsch@alum.mit.edu; jhow@mit.edu).

A. Richards is with the Department of Aerospace Engineering, University of Bristol, Bristol B58 1TR, U.K. (e-mail: arthur.richards@bristol.ac.uk).

Digital Object Identifier 10.1109/TCST.2007.899152

for a real-time implementation over a realistic communication network. This paper also presents a generalization of the implementation architecture for widely separated teams of vehicles. In particular, we define a local neighborhood of each vehicle to be all other vehicles that could have a direct conflict with that vehicle. By limiting the number of vehicles considered to only those within a local region of each vehicle, the number of constraints in each subproblem can be significantly reduced. This modification further simplifies the DRSBK computation, but although the plans are only communicated locally, DRSBK is shown to maintain the robust feasibility of the entire fleet. This architecture generalizes the rigid implementation approaches of [18], [23] to enable some of the vehicles to compute their plans simultaneously, which can significantly reduce the delay incurred.

This paper is organized as follows. Following the problem setup in Section II, Section III presents the RSBK algorithm. Section IV extends the RSBK algorithm to the distributed computation using only local information. Section V shows several simulation results and Section VI shows experimental results on the hardware testbed.

II. Problem Statement

The problem of interest has the overall goal of *reaching the target* while robustly *maintaining feasibility*. In this paper, p, q, r that are used as an index or superscript denote the vehicle number, subscript k denotes the current time step, and subscript j denotes the prediction step. There are total of n vehicles whose dynamics are decoupled and are described by an LTI model

$$\boldsymbol{x}_{k+1}^p = \boldsymbol{A}^p \boldsymbol{x}_k^p + \boldsymbol{B}^p \boldsymbol{u}_k^p + \boldsymbol{w}_k^p \tag{1}$$

for $p = 1, \ldots, n$, where $\boldsymbol{x}_k^p$ is the state vector, $\boldsymbol{u}_k^p$ is the input vector, and $\boldsymbol{w}_k^p$ is the disturbance vector for the pth vehicle. The disturbances $\boldsymbol{w}_k^p$ are unknown but are assumed to lie in known bounded sets

$$\boldsymbol{w}_k^p \in \mathcal{W}^p. \tag{2}$$

The environment has obstacles to be avoided and the vehicles have flight envelope limitations. The general output sets $\mathcal{Y}^p$ capture these local constraints of each vehicle $p = 1, \ldots, n$

$$\boldsymbol{C}^p \boldsymbol{x}_k^p + \boldsymbol{D}^p \boldsymbol{u}_k^p \triangleq \boldsymbol{y}_k^p \in \mathcal{Y}^p. \tag{3}$$

Vehicles are coupled through the constraints and a further set of constraints $c = 1, \ldots, n_c$ are applied to the sum of the outputs from each vehicle

$$\begin{aligned} \forall c: \quad & \boldsymbol{z}_{k,c}^p = \boldsymbol{E}_c^p \boldsymbol{x}_k^p, \quad \forall p = 1, \ldots, n \\ & \sum_{p=1}^{n} \boldsymbol{z}_{k,c}^p \in \mathcal{Z}_c \end{aligned} \tag{4}$$

where $\boldsymbol{z}_{k,c}^p$ denotes p's variable that is coupled with other vehicles' variables. For pair-wise collision avoidance constraints, each constraint c has only two nonzero matrices $\boldsymbol{E}_c^p$ and $\boldsymbol{E}_c^q$, and enforces a minimum separation between that pair of vehicles

$$\|\boldsymbol{r}_k^p - \boldsymbol{r}_k^q\| \geq 2d \tag{5}$$

where $\boldsymbol{r}_k^p$ is a position of the vehicle p, and $2d$ is the minimum separation distance. Note that each set $\mathcal{Z}_c$ is nonconvex in this case. Finally, the objective of the trajectory optimization is to navigate the vehicles to their assigned targets, and the objective function is the sum of individual costs

$$\boldsymbol{x}_{N^p}^p \in \mathcal{X}_{\mathrm{T}}^p \tag{6}$$

$$J = \sum_{p=1}^{n} \sum_{k=0}^{N^p - 1} l^p \left(\boldsymbol{x}_k^p, \boldsymbol{u}_k^p\right) \tag{7}$$

where N^p is the time of arrival at vehicle p's target $\mathcal{X}_{\mathrm{T}}^p$ and is a variable to be minimized, and l^p is a staged cost of vehicle p.

III. Robust Safe But Knowledgeable Algorithm

This section presents a RSBK algorithm [26]. When applied to multi-vehicle control, this algorithm solves a centralized problem, i.e., solving for the plans of all vehicles $p = 1, \ldots, n$ in a single optimization. Section IV discusses how this computation can be separated into a sequence of smaller problems and distributed across the vehicles in the team.

A. Algorithm Description

Solving a single optimization (1)–(7) is not tractable when the vehicle flies through complex environment to a distant target, because the complexity of the optimization grows rapidly with the number of steps N^p required to reach the target. Furthermore, the situational awareness can change as the vehicle flies and there could be significant uncertainties in the far future. It is inefficient to devote considerable computational effort to plans for the far future, since these are likely to be revised in the light of future learning.

The RSBK algorithm does not require the target arrival constraint (6) be satisfied in the planning horizon, allowing the controller to use a short planning horizon. A cost-to-go function is used to provide a good estimate of the remainder of the path to the target, even in a complicated environment [2]. In order to maintain safety [27] of the vehicle under the changes in the environment, the trajectory is required to terminate in a robust control invariant set. However, this set need not be at or around the target, as is common in other MPC methods [8], [9], [18]. More detailed explanation is given later in this subsection.

In this paper, the prediction of a value at time $(k+j)$ made at time k is denoted by the subscript $(k+j \mid k)$. The online MPC optimization develops the control inputs for a short horizon of N steps. The optimization $\mathsf{P}(\boldsymbol{x}_k)$ at time k is defined as

$$J^* = \min_{\substack{\boldsymbol{u}_{\cdot \mid k}^p, \mathcal{Q}_k^p \\ p=1,\ldots,n}} \sum_{p=1}^{n} \left\{ \sum_{j=0}^{N-1} l^p \left(\boldsymbol{u}_{k+j \mid k}^p, \boldsymbol{x}_{k+j \mid k}^p\right) + f^p \left(\boldsymbol{x}_{k+N \mid k}^p\right) \right\} \tag{8}$$

subject to $\forall p = 1, \ldots, n$, and $\forall j = 0, \ldots, N-1$

$$\boldsymbol{x}_{k \mid k}^p = \boldsymbol{x}_k^p \tag{9}$$

$$\boldsymbol{x}_{k+j+1 \mid k}^p = \boldsymbol{A}^p \boldsymbol{x}_{k+j \mid k}^p + \boldsymbol{B}^p \boldsymbol{u}_{k+j \mid k}^p \tag{10}$$

$$\boldsymbol{y}_{k+j \mid k}^p = \boldsymbol{C}^p \boldsymbol{x}_{k+j \mid k}^p + \boldsymbol{D}^p \boldsymbol{u}_{k+j \mid k}^p \in \mathcal{Y}_j^p \tag{11}$$

$$z_{k+j \mid k,c}^p = \boldsymbol{E}_c^p \boldsymbol{x}_{k+j \mid k}^p \tag{12}$$

$$\sum_{p=1}^{n} \boldsymbol{z}^p_{k+j\,|\,k,c} \in \mathcal{Z}_{j,c}, \quad \forall c = 1,\dots,n_c \tag{13}$$

$$\boldsymbol{x}^p_{k+N\,|\,k} \in \mathcal{Q}^p_k \tag{14}$$

$$\mathcal{Q}^p_k = \mathcal{R}^p_k \sim \boldsymbol{L}^p_{N-1}\mathcal{W}^p \tag{15}$$

$$\forall \boldsymbol{x}^p \in \mathcal{R}^p_k$$
$$\Rightarrow \begin{cases} \boldsymbol{A}^p\boldsymbol{x}^p + \boldsymbol{B}^p\kappa^p(\boldsymbol{x}^p) + \boldsymbol{L}^p_{N-1}\boldsymbol{w}^p \in \mathcal{R}^p_k, \quad \forall \boldsymbol{w}^p \in \mathcal{W}^p \\ \boldsymbol{C}^p\boldsymbol{x}^p + \boldsymbol{D}^p\kappa^p(\boldsymbol{x}^p) \in \mathcal{Y}^p_{N-1} \\ \forall c = 1,\dots,n_c: \quad \sum_{p=1}^{n}\boldsymbol{E}^p_c\boldsymbol{x}^p \in \mathcal{Z}_{N-1,c} \\ \qquad \forall(\boldsymbol{x}^1,\dots,\boldsymbol{x}^n) \in \left\{\mathcal{Q}^1_k \times \dots \times \mathcal{Q}^n_k\right\}. \end{cases} \tag{16}$$

The states $\boldsymbol{x}^p_k$ in (9) are the measured states of vehicle p. The decision variables are the control inputs $\boldsymbol{u}^p_{\cdot\,|\,k}$ and the terminal invariant set $\mathcal{Q}^p_k$ that ensures the safety of the vehicle beyond the planning horizon. Note that predictions (10) are made using only the nominal system model, with no disturbance. In order to guarantee robustness against disturbances $\boldsymbol{w}$, the sets $\mathcal{Y}^p_j$ are constructed by tightening the original set $\mathcal{Y}^p$ using a linear controller K^p_j that rejects the disturbance [17]

$$\mathcal{Y}^p_0 = \mathcal{Y}^p, \quad \mathcal{Y}^p_{j+1} = \mathcal{Y}^p_j \sim \left(\boldsymbol{C}^p + \boldsymbol{D}^p\boldsymbol{K}^p_j\right)\boldsymbol{L}^p_j\mathcal{W}^p \quad \forall j^- \tag{17}$$

where the operator $\sim$ denotes the Pontryagin difference that has the property [28]

$$\boldsymbol{a} \in (\mathcal{A} \sim \mathcal{B}), \quad \boldsymbol{b} \in \mathcal{B} \Rightarrow \boldsymbol{a} + \boldsymbol{b} \in \mathcal{A}$$

and L^p_j is a state transition matrix

$$\boldsymbol{L}^p_0 = \boldsymbol{I}, \quad \boldsymbol{L}^p_{j+1} = \left(\boldsymbol{A}^p + \boldsymbol{B}^p\boldsymbol{K}^p_j\right)\boldsymbol{L}^p_j \quad \forall j^-. \tag{18}$$

The notation $\forall j^-$ implies $\forall j = 0,\dots,N-2$, and $\forall j$ implies $j = 0,\dots,N-1$. Equations (12) and (13) represent the inter-vehicle constraints such as collision avoidance, and more details on the implementation are found in Appendix B. Similar tightening is performed on the coupling constraint sets in (13), allowing uncertainty margin for all subsystems within each constraint ($\forall c = 1,\dots,n_c$)

$$\begin{aligned} \mathcal{Z}_{j+1,c} &= \mathcal{Z}_{j,c} \sim \left(\boldsymbol{E}^1_c\boldsymbol{L}^1_j\mathcal{W}_1 \oplus \dots \oplus \boldsymbol{E}^n_c\boldsymbol{L}^n_j\mathcal{W}_n\right) \quad \forall j^- \\ \mathcal{Z}_{0,c} &= \mathcal{Z}_c \end{aligned} \tag{19}$$

where the operator $\oplus$ denotes the Minkowski summation [28]. Unlike other robust MPC approaches, the constraint tightening approach does not increase the complexity of the problem and is well suited for real-time applications. Another advantage of this approach is that the optimization considers the entire range of vehicle dynamics allowed by the constraints (11)–(13).

The set $\mathcal{Q}^p_k$ in (14) is called a *safety* set, defined by (15). The set $\mathcal{R}^p_k$ is a robust control invariant admissible set [29] that has a property (16). The property states that once the vehicle enters the set $\mathcal{R}^p_k$, the vehicle can remain safe indefinitely, satisfying all the constraints using a predetermined terminal control law $\kappa^p(\boldsymbol{x}^p)$. The vehicle is safe also against any changes in the environment that occur beyond this safety set. This terminal set $\mathcal{Q}^p_k$ moves with the vehicle towards the target and therefore a decision variable in the online optimization, as indicated in (8). The RSBK algorithm parameterizes the invariant set, and by using nilpotent candidate controllers, which gives $L^p_{N-1} = 0$, it can solve for a simple nominal control invariant admissible set [16]. One simple invariant set for fixed-wing aircraft is a loiter circle, or for rotorcraft, any point with zero velocity is invariant [27]. Detailed examples are given in Section V. Note that vehicle q's safety set $\mathcal{Q}^q_k$ can overlap with vehicle p's path to its safety set $\mathcal{Q}^p_k$ without any issues. This is because by the time q reaches $\mathcal{Q}^q_k$, vehicle p has already executed the portion that overlaps with $\mathcal{Q}^q_k$.

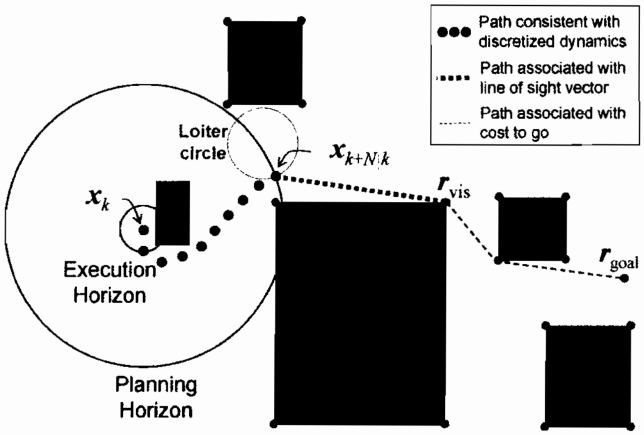

●●● Path consistent with discretized dynamics
▪▪▪▪ Path associated with line of sight vector
...... Path associated with cost to go
Loiter circle
$x_{k+N|k}$
x_k
r_{vis}
r_{goal}
Execution Horizon
Planning Horizon

Fig. 1. Representation of the cost-to-go function showing the three levels of resolution used to approximate a complete path to the goal.

The function $f^p(\boldsymbol{x}^p_{k+N\,|\,k})$ in (8) represents the cost-to-go beyond the planning horizon and is associated with the terminal states of the planned trajectory. The function is designed to provide a reasonable estimate of the length of feasible paths to the goal using a sparse set of grid points, such as obstacle corners [2] shown in Fig. 1 and vertices of other vehicles' safety set. This cost-to-go is based on the current situation awareness of the environment, but because the cost only needs to be evaluated at a small number of points and it uses a coarse model of the aircraft dynamics, it can be updated rapidly as the environment changes [30]. This approach avoids the potential problems of simple terminal penalties, such as the distance to the goal, which are not cognizant of the known obstacles in the environment, and thus can lead to the aircraft becoming trapped behind them.

The implementation of this cost-to-go function has two main components: cost map generation and cost point selection. The cost map generation is based on the observation that the optimized paths (i.e., minimum distance) tend to follow the edges and corners of the obstacles. Thus, a shortest path algorithm is applied to a graph-based representation of the environment to estimate the approximate cost $c^p(\boldsymbol{r}_{\text{corner}})$ to fly from each obstacle corner $\boldsymbol{r}_{\text{corner}}$ to the target of vehicle p. Then, only the pairs of corner locations and costs are stored within the cost map. The receding horizon optimization chooses the best corner $\boldsymbol{r}^p_{\text{vis}}$ that is visible from plan's terminal position $\boldsymbol{r}^p_{k+N\,|\,k}$ of the vehicle p, so that the cost-to-go function is

$$f^p\left(\boldsymbol{x}^p_{k+N\,|\,k}\right) = \left\|\boldsymbol{r}^p_{k+N\,|\,k} - \boldsymbol{r}^p_{\text{vis}}\right\|_2 + c^p\left(\boldsymbol{r}^p_{\text{vis}}\right). \tag{20}$$

Note that the second term gives an estimate of the cost from the selected point to the goal along a collision free path. This cost-to-go function enables the planner to use the environmental information beyond the detailed plan, leading to a knowledgeable trajectory. The discrete decision associated with the corner selection is implemented using mixed integer linear programming (MILP) [2]. The results presented in this paper uses a 2-D version of the cost-to-go calculation, but this has been extended to 3-D in [31] and to account for the typical turning behavior of a UAV in [32].

Given these main components, the overall RSBK algorithm is summarized as follows. From the optimal solution at time k, the first control input $\boldsymbol{u}^{p*}_{k\,|\,k}$ for each vehicle is applied to the system (1). At the next time $k+1$, the states of each vehicle $\boldsymbol{x}^p_{k+1}$ are measured, and the optimization is repeated as follows.

1) Compute the following and store.
   - The constraint sets $\mathcal{Y}^p_j$ through (17) and (18).
   - A cost map $\boldsymbol{r}_{\text{corner}}, c^p(\boldsymbol{r}_{\text{corner}})$ that can be used to evaluate the cost-to-go function $f^p(\cdot)$.
2) At time k, measure vehicle states $\boldsymbol{x}^p_k$, and formulate a MILP problem using the stored values from Step 1). Then, solve the optimization problem $\mathsf{P}(\boldsymbol{x}_k)$ shown with (8)–(14) and (20).
3) Apply control $\boldsymbol{u}^p_k = \boldsymbol{u}^{p*}_{k\,|\,k}$ from the optimal sequence to the system (1).
4) Increment k. If the knowledge of the environment changes, go to Step 1); Otherwise, go to Step 2).

B. Properties

Theorem 3.1 (Robust Feasibility): The RSBK algorithm maintains feasibility of the optimization while satisfying all of the constraints under the action of a bounded disturbance (2), if the first optimization is feasible.

Proof: It can be shown (see Appendix A) that feasibility at time k ensures that a particular candidate solution

$$\hat{\boldsymbol{u}}^p_{k+j+1\,|\,k+1} = \boldsymbol{u}^{p*}_{k+j+1\,|\,k} + K^p_j L^p_j \boldsymbol{w}^p_k, \quad \forall j^- \tag{21}$$

$$\hat{\boldsymbol{u}}^p_{k+j+1\,|\,k+1} = \boldsymbol{u}^{p*}_{k+j+1\,|\,k} + L^p_j \boldsymbol{w}^p_k, \quad \forall j \tag{22}$$

$$\hat{\boldsymbol{u}}^p_{k+N\,|\,k+1} = \kappa^p \left(\hat{\boldsymbol{x}}^p_{k+N\,|\,k+1}\right) \tag{23}$$

$$\hat{\boldsymbol{u}}_{k+N+1\,|\,k+1}p = \boldsymbol{A}^p \hat{\boldsymbol{u}}^p_{k+N\,|\,k+1} + \boldsymbol{B}^p \hat{\boldsymbol{x}}^p_{k+N\,|\,k+1} \tag{24}$$

is feasible at time $k+1$, and hence the optimization at time $k+1$ must be feasible. ■

Remark 1: In order to recursively prove robust feasibility, the algorithm requires the existence of an initial feasible solution. Because the algorithm uses a short planning horizon and does not require the vehicles reach the goal in the first plan, it is typically very easy to find an initial feasible solution, as will be shown in Experimental Results, Section VI. One such initialization is a simple loiter pattern, assuming the vehicles are far enough apart compared to the diameter of the loiter circle. This initialization is much simpler than that required in previous robust multivehicle MPC algorithms [16]. This feature will also be exploited in the distributed form of the algorithm, where initialization can be a significant challenge.

Remark 2: In contrast to the nominal safety approach [27] that assumes no disturbance (i.e., $\mathcal{W}^p = 0$), the algorithm presented here never fails to find a feasible solution under the action of bounded disturbances. Furthermore, the number of control variables is the same as the nominal algorithm. By overbounding the Pontryagin difference operation in (15), (17), and (19), the algorithm will have the same number of constraints [18].

Remark 3: The RSBK algorithm is an *anytime algorithm*, that is, the optimization can be stopped at anytime. In such a case, however, a feasible solution is always available. This follows because a candidate feasible solution can be always constructed from the previous feasible (not necessarily optimal) solution. As shown in Appendix A, the calculation of a candidate solution is simple and involves 1) shifting the previous plan by one time step, 2) adding a disturbance feedback sequence, and 3) appending a terminal control input using κ^p at the terminal step of the plan.

IV. Distributed RSBK Algorithm

This section presents a distributed version of the RSBK algorithm. In this approach, each vehicle solves a reduced subproblem to determine its control inputs. These optimizations are solved in sequence and the distribution is achieved by having each vehicle exchange its plan information with the other vehicles. A key element of this work is that the vehicles must only exchange information with its neighbors, enabling the local optimization to be based on *local information* [23]. This is important because it reduces the communication requirements and enables the groups to replan faster.

A. Algorithm Description

The basic idea is to include only the vehicles that could have direct conflicts with the vehicle that is planning. Fig. 2 shows an example with three aircraft. Any plan of the vehicle r would not have conflict with p's plan because they are far apart. On the other hand, the vehicle q could have a conflict with p if both p and q generate their plans independently and move towards each other. Therefore, p's optimization must include the intention of q, but the vehicle r could be disregarded.

Before presenting the algorithm, several aspects of the notation are defined. First, define vehicle p's neighbor $\mathcal{I}^p_k$ as an ordered set of vehicles whose plans made at time k could have direct conflicts with p's plan made at time k. For the multivehicle collision avoidance problem, the neighborhood of the vehicle p is defined as all the vehicles within distance $2D$ from vehicle p's position, where D is the maximum plan length with some margin and is given by

$$D = \sum_{k=0}^{N} (v_{\max} - \beta_k)\Delta t + d + \alpha_{N-1} + 2\rho \tag{25}$$

with Δt being the sampling time of the discrete time system, d being the size of the vehicle, ρ being the radius of the loiter circle, α_{N-1} being the margin included for robustness [18], and β_k being the constraint tightening margin for the velocity, whose analytical calculations are given later in (40). The dashed line in Fig. 2 shows the boundary of p's neighborhood. The communication range of the vehicles is assumed to be larger than $2D$. All obstacles within range D from the vehicle are assumed to be known. Note that the neighbor set is a function of time k, because the relative position of the vehicles will change over time.

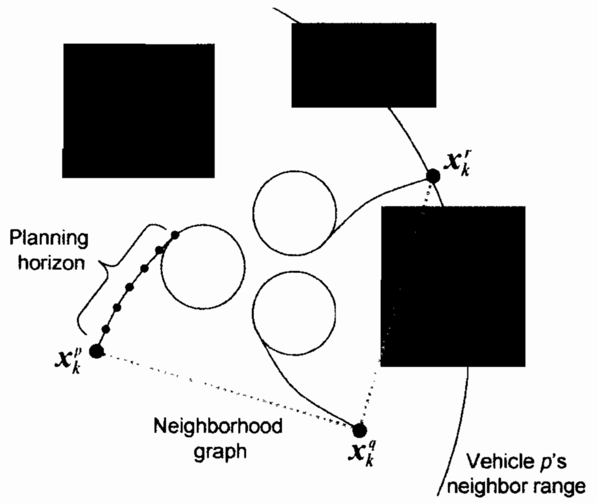


Fig. 2. Neighborhood of each vehicle is shown by the dashed lines. Each plan terminates in a safety circle.

The set $\mathcal{I}_k^p$ also determines the order in which the vehicles calculate their new plans sequentially, although Section IV-E modifies the assumption on this strict ordering. Let $\texttt{pre}(q)$ denote the vehicle ordered prior to the vehicle q, and $\texttt{pre}(q)$ denote the vehicle ordered after q. The first and the last element of this set is expressed as $\texttt{first}(\mathcal{I}_k^p)$ and $\texttt{pre}(\mathcal{I}_k^p)$, respectively. With the definition of $\mathcal{I}_k^p$, we know *a priori* that for any two vehicles p and q with $q \notin \mathcal{I}_k^p$ (and hence $p \notin \mathcal{I}_k^q$), the coupling avoidance constraints are satisfied for all steps with the plans of the two vehicles. Hence, even if the subproblem for the vehicle p includes only the plans of its neighbors, all of the coupling avoidance constraints will be satisfied.

The result of this analysis is that only a subset of all m coupling constraints need to be considered in each subproblem. Define $\mathcal{C}_k^p \subset \{1, \dots, n_c\}$ as the set of coupling constraints to be included in subproblem p at time k. Then, $\mathcal{C}_k^p = \{c \in 1, \dots, n_c : \exists q \in \mathcal{I}_k^p, [\boldsymbol{E}_c^p\ \boldsymbol{E}_c^q] \neq \boldsymbol{0}\}$. This excludes two kinds of constraint irrelevant to p: those that couple p to the vehicles outside its neighborhood and those that do not involve p at all, i.e., with $\boldsymbol{E}_c^p = \boldsymbol{0}$.

Let $\mathcal{G}_k$ denote a vehicle graph whose node is a vehicle and edge connects two nodes if the corresponding vehicles are neighbors. If $\mathcal{G}_k$ is a disconnected graph, then $\mathcal{G}_k$ is divided into a set of connected subgraphs. The information of the neighbor sets $\mathcal{I}_k^p$ is shared by the vehicles in the connected graph (or subgraph if $\mathcal{G}_k$ is not connected), so that the vehicles in the connected graph have the consistent information on the planning order. This can be done using only the inter-vehicle communication. Note that the vehicles that belong to different connected graphs do not need to exchange information because there will be no conflict among them.

B. Algorithm

At time k, the pth vehicle generates its own control inputs $\boldsymbol{u}_{\cdot\,|\,k}^p$ by solving the following optimization subproblem $\mathsf{P}^p(\boldsymbol{x}_k^p)$:

$$\min_{\boldsymbol{u}^p(\cdot\,|\,k), \mathcal{Q}_k^p} \sum_{j=0}^{N-1} l^p\left(\boldsymbol{x}_{k+j\,|\,k}^p, \boldsymbol{u}_{k+j\,|\,k}^p\right) + f^p\left(\boldsymbol{x}_{k+N\,|\,k}^p\right) \quad (26)$$

$$\text{subject to} \quad \forall j: \quad \text{Eq. (9)–(12), (14)}$$

$$\boldsymbol{z}_{k+j\,|\,k,c}^p + \tilde{\boldsymbol{z}}_{k+j\,|\,k,c}^p \in \mathcal{Z}_{j,c}^p, \quad \forall c \in \mathcal{C}_k^p \quad (27)$$

$$\forall \boldsymbol{x}^p \in \mathcal{Q}_k^p \Rightarrow \begin{cases} \boldsymbol{A}^p\boldsymbol{x}^p + \boldsymbol{B}^p\kappa^p(\boldsymbol{x}^p) \in \mathcal{Q}_k^p \\ \boldsymbol{C}^p\boldsymbol{x}^p + \boldsymbol{D}^p\kappa^p(\boldsymbol{x}^p) \in \mathcal{Y}_{N-1}^p \\ \forall c \in \mathcal{C}_k^p : \sum_{q \in \mathcal{I}_k^p} \boldsymbol{E}_c^q\boldsymbol{x}^q \in \mathcal{Z}_{N-1,c} \\ \forall\left(\boldsymbol{x}^{p_0}, \dots, \boldsymbol{x}^{p_n}\right) \in \left\{\mathcal{Q}_k^{p_0} \times \cdots \times \mathcal{Q}_k^{p_n}\right\}. \end{cases} \quad (28)$$

The term $\tilde{\boldsymbol{z}}_{k+j\,|\,k,c}^p$ is a summation of the outputs from the neighbor vehicles and is constant in this local optimization. The term has two components $\forall j, \forall c \in \mathcal{C}_k^p$

$$\tilde{\boldsymbol{z}}_{k+j\,|\,k,c}^p = \sum_{\substack{q \in \mathcal{I}_k^p, \\ \texttt{ord}(q) < \texttt{ord}(p)}} \boldsymbol{z}_{k+j\,|\,k,c}^{q*} + \sum_{\substack{q \in \mathcal{I}_k^p, \\ \texttt{ord}(q) > \texttt{ord}(p)}} \boldsymbol{z}_{k+j\,|\,k-1,c}^{q*}. \quad (29)$$

The first term is the summation over the vehicles that have already planned at time k. The second term is for the vehicles that have not planned at time k, so that the prediction made at $(k-1)$ is used. This prediction comes directly from (12) in the optimization $\mathsf{P}^q(\boldsymbol{x}_{k-1}^q)$. The original coupling constraint sets $\mathcal{Z}_c$ are modified in the following manner, dividing the tightening process from (19) into intermediate stages for each vehicle:

$$\mathcal{Z}_{0,c}^{p_n} = \mathcal{Z}_c \quad (30a)$$

$$\mathcal{Z}_{j,c}^{\texttt{pre}(q)} = \mathcal{Z}_{j,c}^q \sim \boldsymbol{E}_c^q\boldsymbol{L}_j^q\mathcal{W}^q\,\forall j, \quad q \in \mathcal{I}_k^p, \quad q \neq p_0 \quad (30b)$$

$$\mathcal{Z}_{j+1,c}^{p_n} = \mathcal{Z}_{j,c}^{p_0} \sim \boldsymbol{E}_c^{p_0}\boldsymbol{L}_j^{p_0}\mathcal{W}_{p_0}, \quad \forall j^- \quad (30c)$$

with $p_0 = \texttt{first}(\mathcal{I}_k^p)$ and $p_n = \texttt{last}(\mathcal{I}_k^p)$. (30b) tightens the constraints from the vehicle q to $\texttt{pre}(q)$. This represents that the vehicle $\texttt{pre}(q)$ saves some margin for the vehicle q so that q can use it to reject the disturbances $\mathcal{W}^q$. (30c) tightens the constraints from the prediction step j to $(j+1)$. This represents that the optimization at time k for vehicle p_n saves some margin so that the optimization at time $(k+1)$ for vehicle p_0 can use it to also reject the disturbances.

Note that each vehicle uses a nilpotent controller K^p that gives $\boldsymbol{L}_{N-1}^p = 0$, and (15) is not included. For the vehicles that have already planned at time k, the latest solution $\mathcal{Q}_k^q$ is used. For the vehicles that have not planned $\forall q \in \{\texttt{next}(p), \dots, p_n\}$, the invariant set constructed at the previous step is used, i.e., $\mathcal{Q}_k^q = \mathcal{Q}_{k-1}^q$.

The full DRSBK algorithm is as follows.

1) Find a feasible solution of the DRSBK optimization starting from the current states (see Remark 4).
2) Set $k = 1$.
3) For each vehicle p, update the neighbor set $\mathcal{I}_k^p$.
4) For each vehicle p, in a predetermined order (e.g., $1, \dots, n$), do the following.
   a) Gather, by communication, the latest plans $\boldsymbol{z}_{\cdot\,|\,k,c}^{q*}$ or $\boldsymbol{z}_{\cdot\,|\,k-1,c}^{q*}$ from its neighbors $q \in \mathcal{I}_k^p$.
   b) Measure vehicle states $\boldsymbol{x}_k^p$.
   c) Construct a cost map $\boldsymbol{r}_{\text{corner}}, c^p(\boldsymbol{r}_{\text{corner}})$.
   d) Solve subproblem $\mathsf{P}^p(\boldsymbol{x}_k^p)$.
5) Apply control $\boldsymbol{u}_{k\,|\,k}^{p*}$ to each vehicle p.
6) Increment $k := k + 1$ and go to 3.

Note that this algorithm is also a generalization of the two previously published distributed MPC algorithms [18], [23], in that it includes both robustness and a short plan that does not necessarily reach the target.

The steps 1, 3, and 4a–c are implemented in MATLAB. Before solving each subproblem 4d, MATLAB forms the MILP constraints using both the static parameters such as vehicle dynamics limit, target location, and constraint tightening margin, and the dynamically updated parameters such as current vehicle states, obstacle boundaries, and other vehicles' plans. More details on the MILP implementation are shown in Appendix B, Then, the MILP solver CPLEX is invoked in step 4d. The optimized control states and inputs are extracted from CPLEX into MATLAB and is sent to the vehicle in step 5. Note that in the hardware experiment Section VI, MATLAB receives the measured states from the vehicle in step 4b.

C. Robust Feasibility

Even though each subproblem only uses local information, the robust feasibility of the entire fleet can be proven using an approach that parallels [17].

Theorem 4.1: If feasible solutions to all subproblems $\mathsf{P}^1(\boldsymbol{x}_k^1), \ldots, \mathsf{P}^n(\boldsymbol{x}_k^n)$ can be found at time k, then the subproblems at the future times $t > k$ are feasible under the action of disturbances.

Proof: The proof is based on a recursion and similar to the proof of Theorem 3.1 in Appendix A. Without loss of generality, the planning order is assumed to be $1, 2, \ldots, n$.

1) Assume all the subproblems $\mathsf{P}^p(\boldsymbol{x}_k^p)$ have a feasible solution at time k.
2) Then, it can be shown that a feasible solution exists to the first subproblem $\mathsf{P}^1(\boldsymbol{x}_{k+1}^1)$ at time $k+1$ for all disturbances $\boldsymbol{w}_k^1$ acting on the vehicle 1 despite the change in the neighbor set $\mathcal{I}_{k+1}^1$. This is done by showing that the following candidate solution is feasible

$$\hat{\boldsymbol{u}}_{k+j+1\,|\,k+1}^1 = \boldsymbol{u}_{k+j+1\,|\,k}^1 + \boldsymbol{K}_j^1 \boldsymbol{L}_j^1 \boldsymbol{w}_k^{1*} \quad \forall j^- \tag{31}$$

$$\hat{\boldsymbol{x}}_{k+j+1\,|\,k+1}^1 = \boldsymbol{x}_{k+j+1\,|\,k}^{1*} + \boldsymbol{L}_j^1 \boldsymbol{w}_k^1 \quad \forall j \tag{32}$$

$$\hat{\boldsymbol{u}}_{k+N\,|\,k+1}^1 = \kappa^1 \left(\hat{\boldsymbol{x}}_{k+N\,|\,k+1}^1 \right) \tag{33}$$

$$\hat{\boldsymbol{x}}_{k+N+1\,|\,k+1}^1 = \boldsymbol{A}^1 \hat{\boldsymbol{x}}^1)_{k+N\,|\,k+1} + \boldsymbol{B}^1 \hat{\boldsymbol{u}}_{k+N\,|\,k+1}^1. \tag{34}$$

3) Under the assumption in Step 1, it can be shown that given any solution to the problem $\mathsf{P}^p(\boldsymbol{x}_{k+1}^p)$ for $p \in \{1, \ldots, n-1\}$, the next subproblem $\mathsf{P}^{p+1}(\boldsymbol{x}_{k+1}^{p+1})$ is feasible, by showing the feasibility of a candidate sequence. Similar to (31)–(34) in Step 2, the candidate solution is constructed by shifting the previous plan for vehicle $p+1$, assumed known in Step 1, by one time step and adding a perturbation sequence using the predetermined controller $\boldsymbol{K}^{p+1}$.

Therefore, at $k+1$, all subproblems $\mathsf{P}^1(\boldsymbol{x}_{k+1}^1), \ldots, \mathsf{P}^n(\boldsymbol{x}_{k+1}^n)$ are feasible. ■

D. Remarks

Remark 4: Simple Initialization: Initializing this algorithm requires the other vehicles' previous solution, as shown in (28) and (29). However, a simple initialization technique such as loiter circle can be used, as discussed in Remark 1 of the RSBK algorithm.

Remark 5: Scalability: If each subproblem includes the interactions with all the other vehicles, as in [18], the number of constraints grows rapidly with the size of the fleet, which would increase the problem complexity. The algorithm presented here only requires the information about its neighbors, resulting in a more scalable approach. Furthermore, each vehicle only needs the information from its neighbors, so that the algorithm requires much less communication bandwidth.

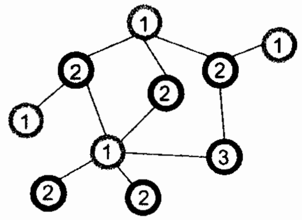


Fig. 3. Output of Brelaz's heuristic algorithm for vertex coloring. Each node represents a vehicle, while each line connecting two nodes represents that they are neighborhood. The number is a group label for the vehicle and vehicles with the same group label can compute simultaneously.

E. Simultaneous Computation

This section removes the assumption on the strict ordering and enables simultaneous computation among vehicles.

Theorem 4.2: Two vehicles p and q can generate trajectories simultaneously without causing infeasibility in the algorithm if $p \notin \mathcal{I}^q$ (and hence $q \notin \mathcal{I}^p$).

Proof: By the definition of neighbor $\mathcal{I}^p$ and $\mathcal{I}^q$, the plans for p and q have no conflict. Given an arbitrary vehicle $r(\neq p, q)$, both optimizations by p and q ensure that the same candidate plan similar to (31)–(34) for each vehicle r is feasible. Thus, when p and q calculate simultaneously, the vehicle r has a feasible solution at the next optimization. ■

By applying this theorem to pairs of vehicles in the fleet, it can be shown that more than two vehicles can perform optimization simultaneously. The vehicles that compute simultaneously are grouped together, and the number of vehicles that compute simultaneously is to be maximized. This grouping problem is cast as a vertex coloring problem on the vehicle graph $\mathcal{G}_k$, where each vertex represents a vehicle and vertices are connected if they are neighbors. The goal is to color all the vertices with a minimum number of colors while using different colors for adjacent vertices. Brelaz's heuristic algorithm [33] is used here because it provides good solutions very rapidly. Vehicles of the same color are in one group and can compute their solutions simultaneously.

Fig. 3 shows a simple example where Brelaz's algorithm is applied to a graph of ten vehicles. Note that in order to color the vehicles, the location of all the vehicles in the connected graph must be known. A central ground station can be introduced to run the grouping algorithm and determine the planning order. Alternatively, the vehicles can obtain this information by communicating only locally through neighbors. Then, Steps 3) and 4) of the DRSBK algorithm in Section IV-B are modified to the following.

3) Ground station receives vehicle positions $\boldsymbol{r}_k^p$, runs the grouping algorithm, and determines the planning order. Each vehicle updates the neighbor set $\mathcal{I}_k^p$.

4) For each group, do the following simultaneously for all vehicles p's in the group.

V. Simulation Results

A. Vehicle Model

A point-mass dynamics model is used to approximate translational dynamics of rotorcraft and fixed-wing aircraft

$$\begin{bmatrix} \boldsymbol{r}^p_{k+1} \\ \boldsymbol{v}^p_{k+1} \end{bmatrix} = \boldsymbol{A}^p \begin{bmatrix} \boldsymbol{r}^p_k \\ \boldsymbol{v}^p_k \end{bmatrix} + \boldsymbol{B}^p \boldsymbol{a}^p_k + \boldsymbol{w}^p_k$$

$$\text{with } \boldsymbol{A}^p = \begin{bmatrix} \boldsymbol{I}_2 & \Delta t \boldsymbol{I}_2 \\ \boldsymbol{O}_2 & \boldsymbol{I}_2 \end{bmatrix}$$

$$\boldsymbol{B}^p = \begin{bmatrix} \frac{(\Delta t)^2}{2} \boldsymbol{I}_2 \\ \Delta t \boldsymbol{I}_2 \end{bmatrix}$$

where $\boldsymbol{r}^p, \boldsymbol{v}^p$, and $\boldsymbol{a}^p$ are the position, the velocity, and the acceleration vector, respectively. Matrices $\boldsymbol{I}_2$ and $\boldsymbol{O}_2$ express an identity matrix and a zero matrix of size 2, respectively. The disturbance $\boldsymbol{w}^p_k$ enters through the input acceleration and

$$\boldsymbol{w}^p_k \in \mathcal{W} = \{\boldsymbol{w} \mid \boldsymbol{w} = \boldsymbol{B}^p \boldsymbol{n}, \boldsymbol{n} \in \mathcal{R}^2, \|\boldsymbol{n}\|_\infty \le w_{\max}\}. \quad (35)$$

The local constraints include the obstacle avoidance, the maximum/minimum speed, and the maximum input constraints

$$\boldsymbol{r}^p_k \notin \mathcal{O}$$
$$v_{\min} \le \|\boldsymbol{v}^p_k\|_2 \le v_{\max}$$
$$\|\boldsymbol{a}^p_k\|_2 \le a_{\max}$$

where $\mathcal{O} \subset \mathcal{R}^2$ expresses the no-fly zones, and $v_{\min}$, $v_{\max}$, $a_{\max}$ are the minimum speed, maximum speed, and maximum acceleration of the vehicle. A two-step nilpotent controller $\boldsymbol{K}^p$ for this system is $\boldsymbol{K}^p = [-(1)/(\Delta t^2) I_2, -(3)/(2\Delta t) I_2]$, which enables the use of *nominal* invariant set as a safety set. Performing constraint tightening (17) gives the following constraint set [18]:

$$\boldsymbol{r}^p_{k+j\,|\,k} \notin \mathcal{O} \oplus \alpha_j \mathcal{B} \quad (36)$$

$$v_{\min} + \beta_j \le \left\|\boldsymbol{v}^p_{k+j\,|\,k}\right\|_2 \le v_{\max} - \beta_j \quad (37)$$

$$\left\|\boldsymbol{a}^p_{k+j\,|\,k}\right\|_2 \le a_{\max} - \gamma_j \quad (38)$$

where constraint contraction parameters α, β, and γ are defined in (40) in Appendix B. The set $\mathcal{B}$ represents a 2-D unit box, i.e., $\mathcal{B} = \{x \in \mathcal{R}^2 \mid \|x\|_\infty \le 1\}$. Note that (36) expands the no-fly zones to guarantee robust feasibility. The cost map calculation is based on the expanded obstacles $\mathcal{O} \oplus \alpha_{N-1}\mathcal{B}$. The inter-vehicle avoidance constraints in p's optimization are written for each $q \in \mathcal{I}^p_k$ as

$$\left\|\boldsymbol{r}^p_{k+j\,|\,k} - \boldsymbol{r}^{q*}_{k+j\,|\,k}\right\| \ge 2d + 2\alpha_j, \quad \text{if } \mathtt{ord}(q) < \mathtt{ord}(p)$$

$$\left\|\boldsymbol{r}^p_{k+j\,|\,k} - \boldsymbol{r}^{q*}_{k+j\,|\,k-1}\right\| \ge 2d + \alpha_j + \alpha_{j+1}, \quad \text{if } \mathtt{ord}(q) > \mathtt{ord}(p)$$

where $\boldsymbol{r}^{q*}_{\cdot\,|\,\cdot}$ are sent from p's neighbors. The terminal safety sets $\mathcal{Q}^p_k$ must not overlap with each other, as shown in (28), so that the sets $\mathcal{Q}^q_k$ ($\forall q \neq p$) are treated as no-fly zones after time step $k + N - 1$ in the optimization $\mathsf{P}^p(\boldsymbol{x}^p_k)$. These nonconvex avoidance constraints are implemented using MILP. More details are found in Appendix B.

B. Long Trajectory Generation for Single Vehicle

The first example demonstrates that the RSBK algorithm for a single vehicle can design a very long trajectory without computational issues. In this example, the rotorcraft is used, which does not have the minimum speed constraint. The vehicle parameters are: $\Delta t = 2.6$ s, $N = 6, v_{\max} = 0.5$ m/s, and $a_{\max} = 0.17$ m/s^2. The invariance of the set $\mathcal{Q}_k$ is guaranteed by imposing the following hovering safety constraints in the optimization

$$\boldsymbol{x}_{k+N+1\,|\,k} = \boldsymbol{x}_{k+N\,|\,k}$$
$$\boldsymbol{r}_{k+N\,|\,k} \notin \mathcal{O} \oplus \alpha_{N-1}\mathcal{B}.$$

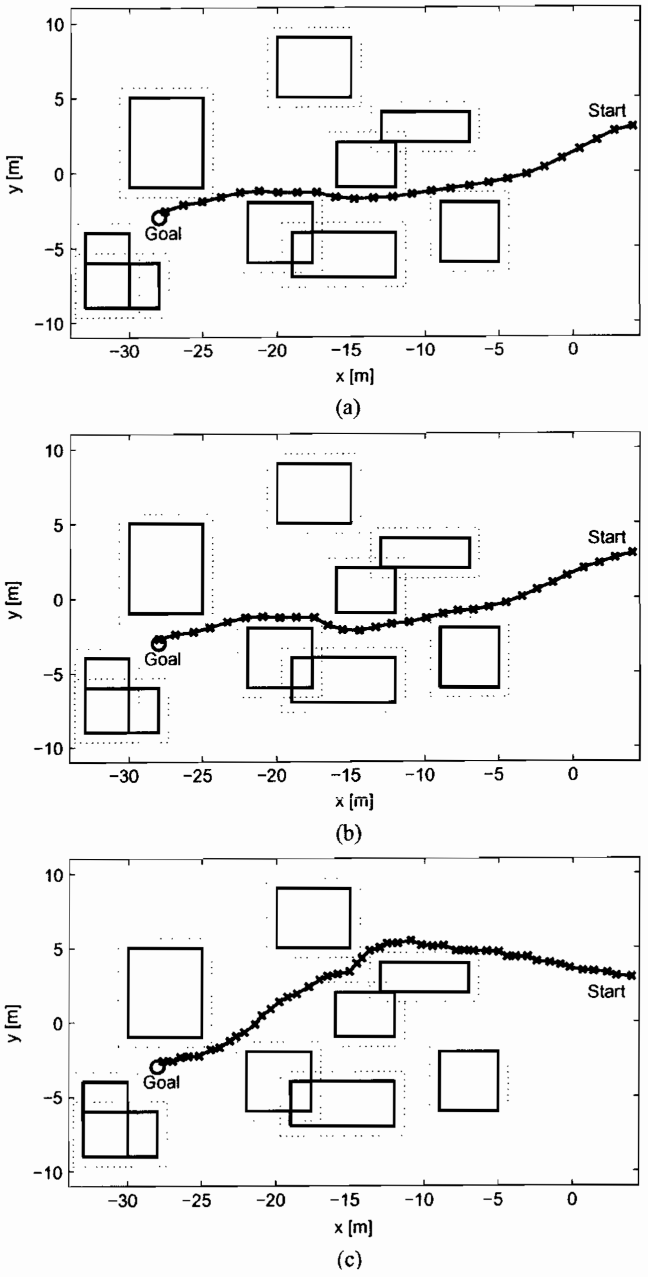
The second equation ensures that the hovering location is collision free. The target region is far from the initial vehicle, and to solve this problem by planning all the way to the goal would require a horizon of at least 30 steps. This would require impractical computational effort for real-time applications and, as discussed earlier, would be inefficient due to the uncertainty in the far future.

Fig. 4 shows trajectories generated under three different disturbance levels $w_{\max} = 0, 0.1a_{\max}, 0.2a_{\max}$. In all cases, the RSBK algorithm guided the vehicle to the target, and the average computation time was less than 0.2 s. When the disturbance level is 10% of the control authority, the trajectory is similar to the one with no disturbance. However, when the disturbance level is raised to 20% of the control authority, the vehicle takes a different route because the passage in the middle of the figure used by the other plans is too narrow to pass through robustly. A cost-to-go calculation based on the robustified environment $\mathcal{O} \oplus \alpha_{N-1}\mathcal{B}$ does not allow the vehicle to enter the narrow passage where the vehicle could violate the collision avoidance constraints due to a strong disturbance.

Note that the vehicle moves slowly when the disturbance is strong, as it is expected intuitively. Because more margin must be saved to reject a stronger disturbance, less control authority can be used when generating the trajectory. The hovering state used as a terminal invariant set requires the vehicle be able to stop at the end of each plan using the small control authority available in the prediction. Table I summarizes this result. The average speed becomes significantly smaller when the disturbance level is increased from 10% to 20%. The number of steps it takes to reach the target set is significantly longer with the 20% disturbance level, partly because of the longer route it chooses, but mainly due to the reduced speed.

C. Multi-UAV Scenarios

The second set of simulations used homogeneous fixed-wing UAVs. The maneuver limit of the vehicle is given by $v_{\min} = 18$ m/s, $v_{\max} = 24$ m/s, $a_{\max} = 3.84$ m/s^2. The disturbance magnitude $w_{\max}$ is 5% of the control authority $a_{\max}$. The planning horizon length N is 5. Fixed-wing UAVs have minimum speed limit, and a safety loitering circle is used as a terminal invariant set [23]. For simplicity, in this section, the simultaneous computation is implemented as a sequential computation on a single computer but in the same simulation time step. Section VI shows the real-time experimental results using multiple computers.



(a)

(b)

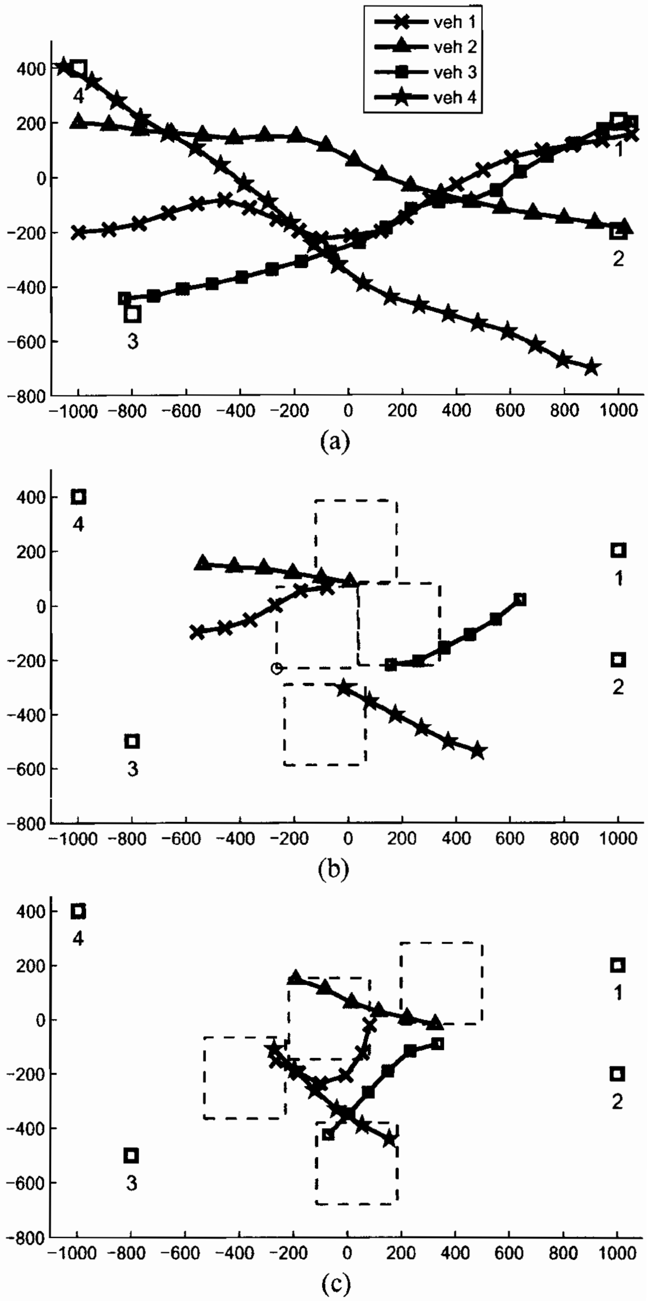
(c)

Fig. 4. Trajectories generated by the RSBK algorithm. The vehicle starts at the right, and the goal is marked with o. The obstacles are expanded in the RSBK calculation to account for the avoidance check imposed only at discrete time steps. (a) No disturbance. (b) Disturbance level 10%. (c) Disturbance level 20%.

TABLE I
PERFORMANCE COMPARISON FOR THREE DISTURBANCE LEVELS

Disturbance level	Average speed	Steps
0 %	0.50 m/s	26
10 %	0.44 m/s	30
20 %	0.28 m/s	48

The DRSBK algorithm was tested in the following two scenarios. The first scenario uses four vehicles with vehicle avoidance constraints. Fig. 5(a) shows the entire trajectories. Goals are marked with □ together with the corresponding vehicle indices. Fig. 5(b) shows the plans made at time $k = 5$. The rectangle in dashed lines shows a safety region where the safety circle is contained and the other vehicles cannot enter after time $k + N$. Note that the plan of the vehicle 4 (marked with a star) aims for the corner (marked with a o) of this rectangle of the vehicle 1 because this corner is in the cost map. As shown in Fig. 5(b) and (c), it is acceptable for the plan of one vehicle to pass through the safety region for another. The terminal set (28) only requires that the safety regions do not overlap each other.



(a)

(b)

(c)

Fig. 5. Four vehicle scenario. The goal points are shown with □ with the corresponding vehicle index. (a) Trajectories. (b) Snapshot at time 5. Note that squares containing safety circles do not overlap with each other. (c) Snapshot at time 8, showing the successful avoidance maneuvers.

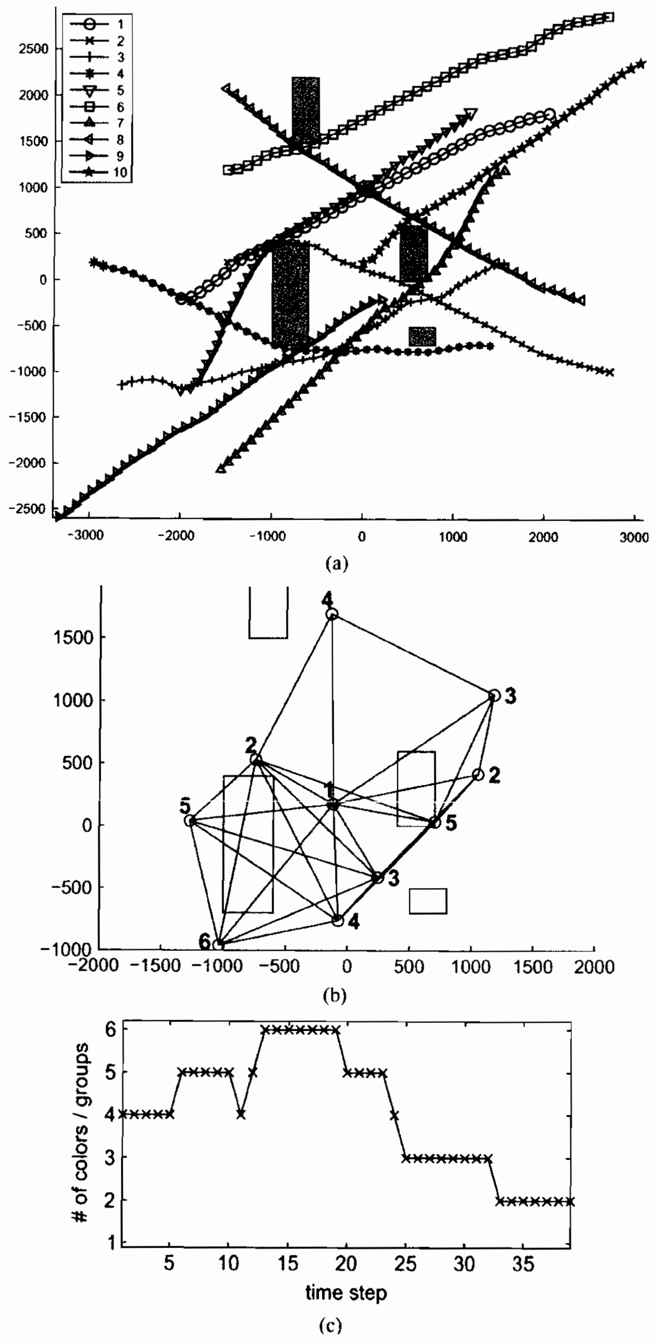


Fig. 6. Ten vehicle scenario with four obstacles. (a) Trajectories of all the vehicles. (b) Graph representation of neighborhood at time 14. (c) Time history of the vehicle grouping.

TABLE II
Average Computation Time (s) of Each Subproblem

Scenario	Cost map calculation	Optimization (MILP)
4 veh	0.04	0.21
10 veh (local comm.)	0.21	0.25
10 veh (full comm.)	0.21	0.37

The second scenario is much more complicated and involves ten vehicles and four obstacles. Fig. 6(a) shows the trajectories of all ten vehicles. Although computation was done on one processor in this section, the grouping algorithm was included to investigate the potential for speed-up by simultaneous computation. The hardware experiments presented in Section VI use one processor for each vehicle. Fig. 6(b) shows a snapshot of the vehicle locations (marked with a ○) at time $k = 14$. The neighbors are connected by the lines and each vehicle is labeled with a color/group number. Note that no two vehicles connected to each other have the same group number. The vehicles in the same group can simultaneously solve their optimization without any conflict in their trajectories. Fig. 6(c) shows the time history of the number of colors required for grouping the vehicles. The number of groups is low when the vehicles are far apart, but as might be expected, this increases to six in the middle of the mission when the vehicles are in close proximity.

Table II shows the average computation time for these scenarios. The cost map calculation was done in MATLAB, and the MILP optimization was solved using CPLEX 9.0 on a Pentium IV 3.2-GHz machine with 1 GB of RAM. The computation time of the cost map calculation grows with the number of vehicles because a higher number of loiter circles means that more obstacles must be considered. In order to demonstrate the effect of using only the local information, the ten-vehicle scenario is tested also with a case where each vehicle includes all other vehicles as neighbors with full communication. The last two rows of Table II illustrate that DRSBK with local communication solves the problem much faster. The output of the grouping algorithm is used to enable simultaneous computation, and the number of groups that must compute sequentially is ten in the full communication case, as opposed to six in the local communication case. This indicates the local communication architecture reduces the fleet computation time further by 40% in this scenario.

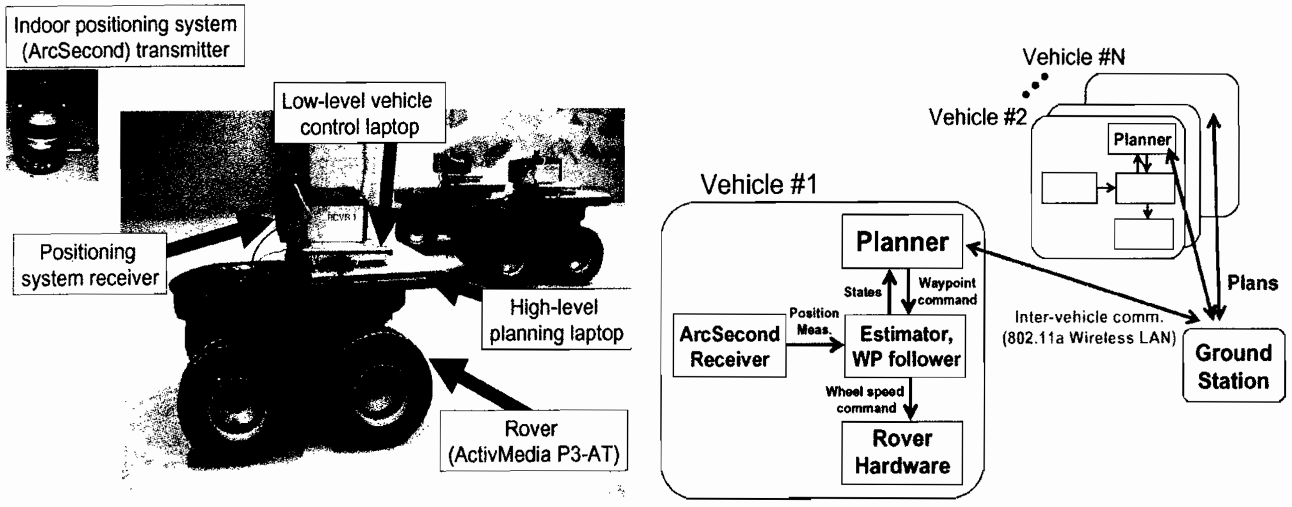
VI. Experimental Results

This section presents experimental results of the DRSBK algorithm on the multivehicle testbed. The hardware demonstrations introduce realistic features such as computation and communication time delays and prediction errors that naturally arise from the various sources of uncertainty in the system, including the tracking errors from the low-level waypoint follower and modeling errors of the vehicle dynamics. These implementation challenges must be addressed by the algorithm in order to successfully generate trajectories online.

A. Testbed Setup

Fig. 7 shows the testbed setup with the indoor positioning system from ArcSecond Constellation 3-D-i and Pioneer 3-AT from ActivMedia Robotics. In order to demonstrate the online distributed computation amongst the vehicles in the fleet, each rover has two laptops, as shown in Fig. 7(a). A small "control" laptop performs the navigation and low-level vehicle control tasks, and a 2.4-GHz "planning" laptop performs the DRSBK computation using a combination of MATLAB and CPLEX.

The control laptop runs an estimator for the position and the velocity estimate of the vehicle. For practical implementation, instead of applying the acceleration command $\boldsymbol{u}^*$ as in Algorithm step 5, the onboard planner sends the optimized trajectory



(a)

(b)

Fig. 7. Multirover testbed. (a) Photos of the hardware. (b) Testbed architecture.

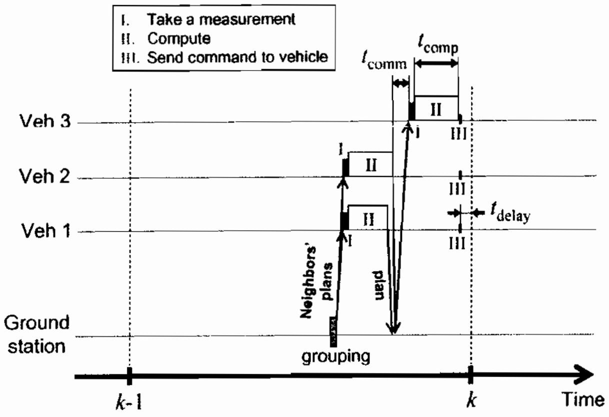


Fig. 8. Timing of the DRSBK algorithm on the testbed.

to the control laptop, which generates wheel speed commands for the rover. A nonlinear guidance law [34] is used to implement a trajectory tracking controller, which runs at a faster rate than the DRSBK controller. This represents an apportionment of uncertainty in the problem, with the low-level handling fast dynamics and the high-level handling uncertainty in the environment, collision avoidance, and residual tracking errors. The ground station laptop runs a grouping algorithm at each time step, but all DRSBK calculations are done onboard, as shown in Fig. 8. Each planning laptop communicates its local solution with its neighbors using the 802.11a wireless LAN. For this testbed, the inter-vehicle communication is facilitated using an access point connected through an Ethernet cable to the ground station laptop.

Fig. 8 shows the timing of the experimental setup. This example has three vehicles in two groups where the vehicles 1 and 2 compute simultaneously. The control input of each vehicle is implemented using fixed discrete time steps. The planner takes a measurement (I) and propagates forward the measured states using the nominal model to predict the initial states $\boldsymbol{x}_k$ of the plan. This propagation compensates for the system delay that results from the computation time t_{comp}, the communication delay t_{comm}, and the actuation delay t_{delay}. It then computes the optimal control input (II) and waits until the control update time (III). The step size Δt between time step k and $k+1$ was 2.8 s for two-rover cases and 3.5 s for three-rover cases.

A typical experimental run starts by commanding the vehicles to drive straight in the initial heading direction. After 1.5 s, the first vehicle takes its measurement and the DRSBK loop starts. For other vehicles that have not made any plans, loiter circles starting from their current states are used as their initial feasible plan, as mentioned in Remark 4. This demonstrates the online initialization capability of this algorithm.

Given the applications of interest are multi-UAV coordination problems, the rovers have been modified to emulate the motion of a UAV in 2-D. In particular, the vehicles are constrained to a maximum speed $v_{\max} = 0.25$ m/s, a minimum speed $v_{\min} = 0.044$ m/s, and a minimum turning radius $r_{\min} = 0.9$ m. The vehicle size is $d = 0.25$ m. The planning horizon length is three steps. The disturbance $\boldsymbol{w}_k^p$ is assumed to enter into position and velocity separately

$$\boldsymbol{w}_k^p \in \mathcal{W} = \{\boldsymbol{w} \in \mathcal{R}^4 \mid \|[I_2, O_2]\boldsymbol{w}\|_2 \le w_{r_{\max}}, \|[O_2, I_2]\boldsymbol{w}\|_2 \le w_{v_{\max}}\}.$$

Extensive testing of the vehicle on different types of flooring indicated that the prediction errors due to the uncertain vehicle dynamics, navigation errors and external disturbances are approximately $w_{r_{\max}} = 15$ cm and $w_{v_{\max}} = 5$ cm/s. Due to the tightened constraints, the speed is constrained to be 0.14 m/s $\le v \le$ 0.15 m/s after $N = 3$ steps.

B. Results

Scenarios are constructed to highlight several features of DRSBK algorithm: on-board laptops generate trajectories online, which shows the computational advantages for real-time

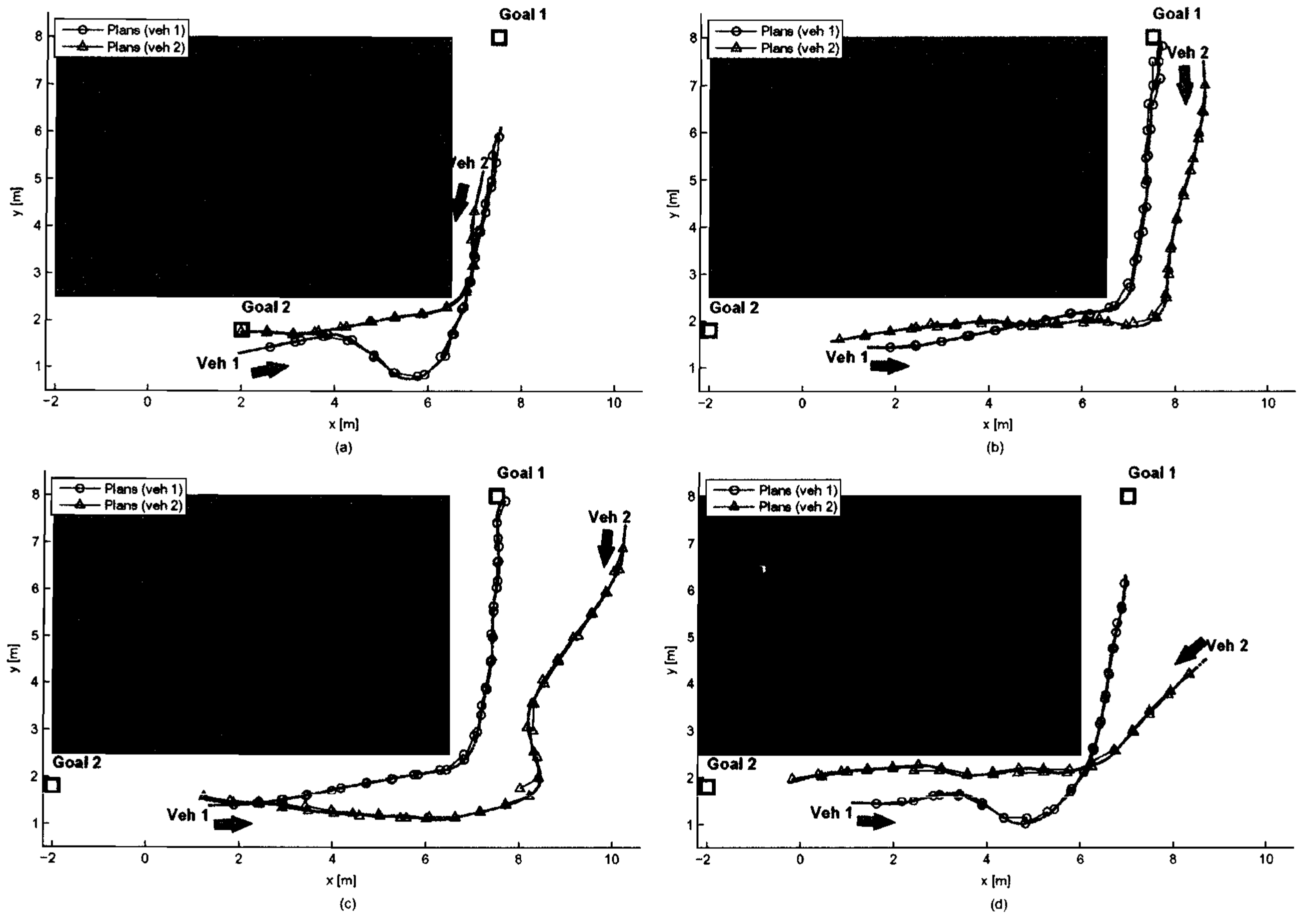


Fig. 9. Two vehicle experiment results. The arrows show the initial heading directions of the vehicles. The goals are marked with a □. (a) Run #1. (b) Run #2. (c) Run #3. (d) Run #4.

applications; the vehicles are required to maneuver in a constrained environment, which demonstrates the robust feasibility under the action of disturbances; plans based on distributed computation can satisfy the coupling collision avoidance constraints.

Test 1: The first experiments were designed to test the obstacle and vehicle avoidance using two rovers. During the first few steps in each run, the separation between the two vehicles was more than 2-D = 6.34 m, and the onboard computers optimize trajectories simultaneously. However, as they move towards each other, the planning horizons overlap, and they must compute the solutions sequentially. For the purposes of these demonstration, the experiment is terminated once the vehicle avoidance and obstacle avoidance maneuvers are completed. Fig. 9 shows four runs performed on this testbed. DRSBK algorithm maintained feasibility under the action of the disturbances, and all runs show the robust vehicle avoidance and obstacle avoidance based on the online distributed trajectory generation.

Test 2: The second set of runs examines vehicle avoidance maneuvers using three rovers that are forced to execute a crossing pattern. Fig. 10 presents the executed trajectories for three runs with different initial locations and headings. Note that the resolution strategies differ with the scenario. One of the key features of MILP is that it handles the nonconvexity directly and looks for solutions on all sides of obstacles and conflicts. This example illustrates that DRSBK is making use of this functionality, as opposed to other methods that could simply refine the initial guess that are given. In these scenarios, vehicles 1 and 3 are initially neighbors because they are closer than $2D = 6.54$ m, and thus compute sequentially. Vehicle 2 is initially independent of that pair, and thus solves for its plan simultaneously with one of them. However, since the vehicles are crossing, vehicle 2 joins the pair after a few time steps, and then all three vehicles compute sequentially. Once the vehicles finish the avoidance maneuver near the middle of the figure, the group breaks up as the vehicles move apart and starts solving for the plans simultaneously again. The results demonstrate online dynamic grouping and regrouping of the vehicles using the algorithm in Section IV-E.

VII. Conclusion

This paper presented a new distributed robust MPC algorithm for multivehicle trajectory optimization. The approach extends previous results to ensure robust feasibility without having to plan all of the way to the goal and with only communicating the plans within a local neighborhood rather than the entire fleet. This two new features greatly reduce the computation effort and facilitate a significantly more general implementation architecture for the distributed trajectory optimization. Experimental re-

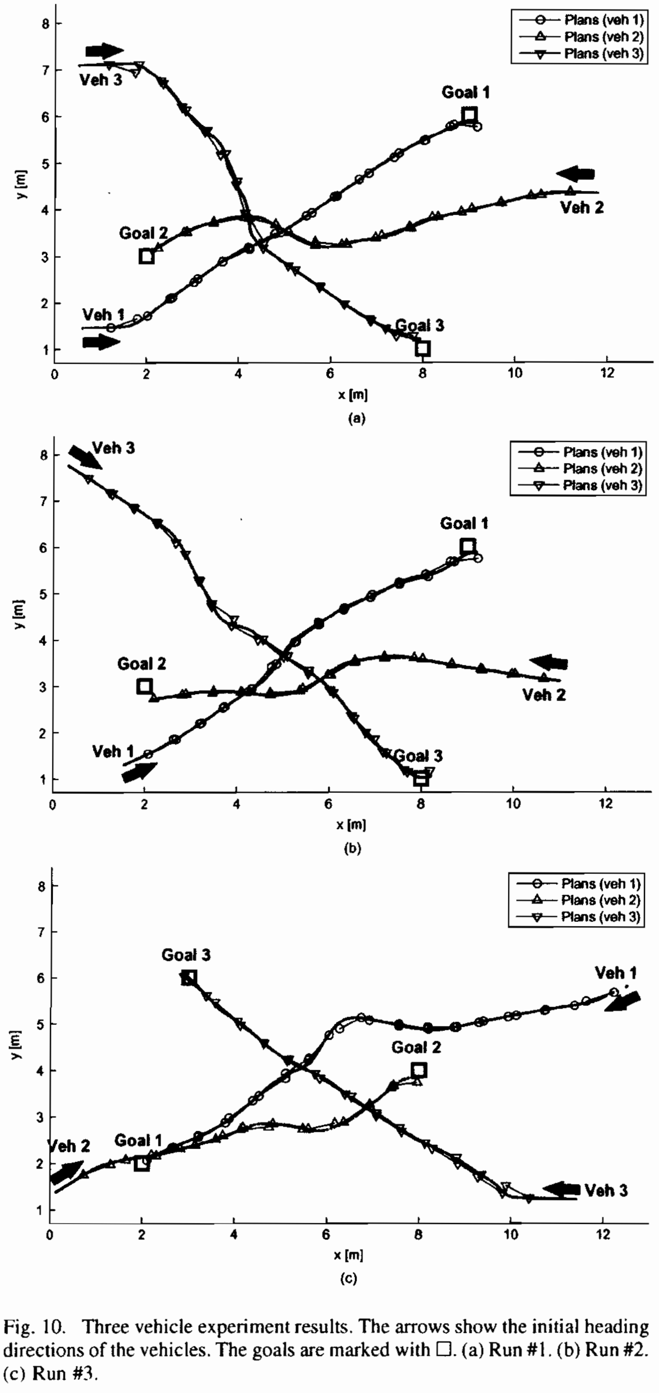


Fig. 10. Three vehicle experiment results. The arrows show the initial heading directions of the vehicles. The goals are marked with □. (a) Run #1. (b) Run #2. (c) Run #3.

sults on a multivehicle testbed demonstrate many advantages of this algorithm including online distributed optimization, simultaneous computation, and the robust feasibility against the disturbances in the real environment.

APPENDIX

A. *Proof of Theorem 3.1*

The proof of Theorem 3.1 is based on a recursion [17]. Once a feasible solution $\boldsymbol{u}^{p*}_{\cdot\,|\,k}, \boldsymbol{x}^{p*}_{\cdot\,|\,k}$ to the problem $\mathsf{P}(\boldsymbol{x}_k)$ is obtained at time k, a candidate solution to $\mathsf{P}(\boldsymbol{x}_{k+1})$ at time $k+1$ is constructed from (21)–(24). Note that the disturbance realization $\boldsymbol{w}^p_k$ at time k is available at time $(k+1)$. Using that the feasible solution $\boldsymbol{u}^{p*}_{\cdot\,|\,k}, \boldsymbol{x}^{p*}_{\cdot\,|\,k}$ satisfies constraints (10)–(14), it can be shown that

$$\begin{aligned}
&\boldsymbol{A}^p\hat{\boldsymbol{x}}^p_{k+j+1\,|\,k+1} + \boldsymbol{B}^p\hat{\boldsymbol{u}}^p_{k+j+1\,|\,k+1}\\
&\quad = \boldsymbol{A}^p\boldsymbol{x}^{p*}_{k+j+1\,|\,k} + \boldsymbol{B}^p\boldsymbol{u}^{p*}_{k+j+1\,|\,k} + (\boldsymbol{A}^p + \boldsymbol{B}^p\boldsymbol{K}^p)\boldsymbol{L}^p_j\boldsymbol{w}^p_k\\
&\quad = \hat{\boldsymbol{x}}^p_{k+j+2\,|\,k+1}, \quad \forall j^-
\end{aligned}$$

satisfying (10) at $k+1$. Similarly, (11) at time $k+1$ is

$$\begin{aligned}
&\boldsymbol{C}^p\hat{\boldsymbol{x}}^p_{k+j+1\,|\,k+1} + \boldsymbol{D}^p\hat{\boldsymbol{u}}^p_{k+j+1\,|\,k+1}\\
&\quad = \boldsymbol{C}^p\boldsymbol{x}^{p*}_{k+j+1\,|\,k} + \boldsymbol{D}^p\boldsymbol{u}^{p*}_{k+j+1\,|\,k} + (\boldsymbol{C}^p + \boldsymbol{D}^p\boldsymbol{K}^p)\boldsymbol{L}^p_j\boldsymbol{w}^p_k\\
&\quad \in \mathcal{Y}^p_j, \quad \forall j^-
\end{aligned}$$

since it can be shown using (17) that

$$\begin{aligned}
&\boldsymbol{y}^{p*}_{k+j+1\,|\,k} \in \mathcal{Y}^p_{j+1}, \quad \boldsymbol{w}^p_k \in \mathcal{W}^p\\
&\Rightarrow \boldsymbol{y}^{p*}_{k+j+1\,|\,k} + (\boldsymbol{C}^p + \boldsymbol{D}^p\boldsymbol{K}^p)\boldsymbol{L}^p_j\boldsymbol{w}^p_k \in \mathcal{Y}^p_j.
\end{aligned}$$

It can be shown using (19) that the coupling constraints (13) at time $k+1$ is satisfied

$$\begin{cases}
\sum_{p=1}^n \boldsymbol{E}^p_c\hat{\boldsymbol{x}}^p_{k+j+1\,|\,k+1} = \sum_{p=1}^n \left(\boldsymbol{E}^p_c\boldsymbol{x}^{p*}_{k+j+1\,|\,k} + \boldsymbol{L}^p_j\boldsymbol{w}^p_k\right)\\
\sum_{p=1}^n \boldsymbol{E}^p_c\boldsymbol{x}^{p*}_{k+j+1\,|\,k} \in \mathcal{Z}_{j+1,c}\\
\boldsymbol{w}^p_k \in \mathcal{W}^p
\end{cases}$$

$$\Rightarrow \sum_{p=1}^n \boldsymbol{E}^p_c\hat{\boldsymbol{x}}^p_{k+j+1\,|\,k+1} \in \mathcal{Z}_{j,c}.$$

For the terminal constraints (14) at $k+1$

$$\begin{aligned}
&\boldsymbol{x}^{p*}_{k+N\,|\,k} \in \mathcal{Q}^p_k = \mathcal{R}^p_k \sim \boldsymbol{L}^p_{N-1}\mathcal{W}^p\\
&\Rightarrow \hat{\boldsymbol{x}}^p_{k+N\,|\,k+1} \in \mathcal{R}^p_k\\
&\Rightarrow \hat{\boldsymbol{x}}^p_{k+N+1\,|\,k+1} + \boldsymbol{L}^p_{N-1}\boldsymbol{w} \in \mathcal{R}^p_k, \quad \forall \boldsymbol{w} \in \mathcal{W}^p\\
&\Rightarrow \hat{\boldsymbol{x}}^p_{k+N+1\,|\,k+1} \in \mathcal{Q}^p_k.
\end{aligned}$$

Last, the following shows the output constraint for the terminal step is satisfied:

$$\begin{aligned}
&\hat{\boldsymbol{x}}^p_{k+N\,|\,k+1} \in \mathcal{R}^p_k\\
&\Rightarrow \hat{\boldsymbol{y}}^p_{k+N+1\,|\,k+1} = \boldsymbol{C}^p\hat{\boldsymbol{x}}^p_{k+N\,|\,k+1}\\
&\qquad\qquad + \boldsymbol{D}^p\kappa^p\left(\hat{\boldsymbol{x}}^p_{k+N\,|\,k+1}\right) \in \mathcal{Y}^p_{N-1}.
\end{aligned}$$

Therefore, the candidate solution satisfies, under the bounded disturbance $\boldsymbol{w}^p_k \in \mathcal{W}^p$, all the output constraints $\mathcal{Y}^p_j$ and coupling constraints $\mathcal{Z}_{j,c}$ at time $k+1$, and the terminal state $\hat{\boldsymbol{x}}^p_{k+N+1\,|\,k+1}$ lies in the set $\mathcal{Q}^p_k$. Thus, feasibility at time k guarantees feasibility at time $k+1$ under the action of bounded disturbance. If the first optimization $\mathsf{P}(\boldsymbol{x}_0)$ is feasible, then all the future optimizations will be feasible. ■

B. *MILP Implementation of DRSBK Online Optimization*

This Appendix shows the detailed MILP implementation of DRSBK algorithm. The disturbance is assumed to be infinity-norm bounded here, i.e., $\mathcal{W}^p = \{G\boldsymbol{w} \mid \|\boldsymbol{w}\|_\infty \leq w_{\max}\}$.

1) Constraint Tightening for Robustness: The constraint tightening in (17) and (30a)–(30c) are implemented using the following constraint contraction parameters [18]:

$$\begin{aligned} \alpha_0 &= 0, \quad \alpha_j = \alpha_{j-1} + \left\| [1\,0\,0\,0] \boldsymbol{L}^p_{j-1} \boldsymbol{B}^p \boldsymbol{G} \right\|_1 w_{\max} \\ \beta_0 &= 0, \quad \beta_j = \beta_{j-1} + C \left\| [0\,0\,1\,0] \boldsymbol{L}^p_{j-1} \boldsymbol{B}^p \boldsymbol{G} \right\|_1 w_{\max} \\ \gamma_0 &= 0, \quad \gamma_j = \gamma_{j-1} + C \left\| [1\,0] \boldsymbol{K}^p_{j-1} \boldsymbol{L}^p_{j-1} \boldsymbol{B}^p \boldsymbol{G} \right\|_1 w_{\max} \\ & \qquad j \geq 1 \end{aligned} \tag{40}$$

where α_j, β_j, and γ_j respectively represents the constraint contraction for position, velocity, and input for the jth prediction step. The coefficient $C = 1$ when the constraint set (37) and (38) and the disturbance set are both two-norm bounded. However, $C = \sqrt{2}$ when performing the Pontryagin difference between a two-norm bounded set and the infinite-norm bounded disturbance set $\mathcal{W}$ in (35). This is because $\mathcal{W}$ has the maximum magnitude of the length $\sqrt{2} w_{\max}$ in the diagonal directions.

2) Output Constraint Set (11): The obstacle avoidance constraints use binary variables. For each point $\boldsymbol{r}^p_{k+j\,|\,k} = [x^p_{k+j\,|\,k}, y^p_{k+j\,|\,k}]^T$ and each rectangular shaped obstacle defined by two corners $[x_{\text{low}}, y_{\text{low}}]^T$ and $[x_{\text{high}}, y_{\text{high}}]^T$, the avoidance constraints can be expressed as

$$\forall o, \forall j: \quad x^p_{k+j\,|\,k} \leq x_{\text{low},_o} - \alpha_j + M b^p_{\text{obst},_{jo1}} \tag{41a}$$

$$y^p_{k+j\,|\,k} \leq y_{\text{low},_o} - \alpha_j + M b^p_{\text{obst},_{jo2}} \tag{41b}$$

$$x^p_{k+j\,|\,k} \geq x_{\text{high},_o} + \alpha_j - M b^p_{\text{obst},_{jo3}} \tag{41c}$$

$$y^p_{k+j\,|\,k} \geq y_{\text{high},_o} + \alpha_j - M b^p_{\text{obst},_{jo4}} \tag{41d}$$

$$\sum_{i=1}^{4} b^p_{\text{obst},_{joi}} \leq 3 \tag{41e}$$

where M is a large number to relax the constraints in (41a)–(41d), and o denotes the index of the obstacle. The logical constraint (41e) requires at least one constraint in (41a)–(41d) be active. Note that the parameter α_j tightens the constraints by enlarging the obstacles.

The output constraint (11) also includes the bound on speed and inputs. Let vectors $\boldsymbol{r}, \boldsymbol{v}$,, and $\boldsymbol{a}$, respectively represent position, velocity, and acceleration input in the inertia frame. A set of n_d linear constraints approximates the two-norm bounded constraints on the acceleration and velocity vectors, which in turn limits the maximum turning rate

$$[\cos\theta_m, \sin\theta_m] \boldsymbol{v}^p_{k+j\,|\,k} \leq v_{\max} - \beta_j \tag{42a}$$

$$[\cos\theta_m, \sin\theta_m] \boldsymbol{a}^p_{k+j\,|\,k} \leq a_{\max} - \gamma_j \tag{42b}$$

$$\theta_m = \tfrac{2\pi m}{n_d}, \quad \forall m = 1, \ldots, n_d.$$

The minimum speed constraint is nonconvex and requires n_v binary variables to express in MILP

$$[\cos\theta_m, \sin\theta_m] \boldsymbol{v}^p_{k+j\,|\,k} \geq v_{\min} + \beta_j - 2 v_{\max} b^p_{\text{vel},_{jm}} \tag{43a}$$

$$\sum_{m=1}^{n_v} b^p_{\text{vel},_{jm}} \leq n_v - 1. \tag{43b}$$

One advantage of MILP is that the optimization can consider the entire range of vehicle dynamics allowed by these constraints.

3) Invariance Constraints (14): From the terminal states, the vehicle has an option to enter a left or right loiter circle. The centers of the left and right safety circles are

$$\boldsymbol{O}^p_{\text{L}} = \boldsymbol{r}^p_{k+N\,|\,k} + R\left(\frac{\pi}{2}\right) \frac{\rho}{v_{\max} - \beta_{N-1}} \boldsymbol{v}^p_{k+N\,|\,k} \tag{44a}$$

$$\boldsymbol{O}^p_{\text{R}} = \boldsymbol{r}^p_{k+N\,|\,k} + R\left(-\frac{\pi}{2}\right) \frac{\rho}{v_{\max} - \beta_{N-1}} \boldsymbol{v}^p_{k+N\,|\,k} \tag{44b}$$

where $R(\theta)$ is a rotation matrix of angle θ, and ρ is the radius of the turning circle given by

$$\rho = \frac{(v_{\max} - \beta_{N-1})^2}{a_{\max} - \gamma_{N-1}} \times \frac{v_{\max} - \beta_{N-1}}{v_{\min} + \beta_{N-1}}.$$

The second term accounts for the variability of the terminal speed $\|\boldsymbol{v}^p_{k+N\,|\,k}\|$. The binary variable b^p_{left} chooses either the left or right safety circle

$$\boldsymbol{O}^p_{\text{L}} - 2(\rho + \alpha_{N-1})(1 - b^p_{\text{left}}) \leq \boldsymbol{O}^p \tag{45a}$$

$$\boldsymbol{O}^p_{\text{L}} + 2(\rho + \alpha_{N-1})(1 - b^p_{\text{left}}) \geq \boldsymbol{O}^p \tag{45b}$$

$$\boldsymbol{O}^p_{\text{R}} - 2(\rho + \alpha_{N-1}) b^p_{\text{left}} \leq \boldsymbol{O}^p \tag{45c}$$

$$\boldsymbol{O}^p_{\text{R}} + 2(\rho + \alpha_{N-1}) b^p_{\text{left}} \geq \boldsymbol{O}^p. \tag{45d}$$

With the notation $\boldsymbol{O}^p = [x^p_{\text{center}}, y^p_{\text{center}}]^{\text{T}}$, the obstacle avoidance constraints of the safety circle are written as

$$\forall o: \quad x^p_{\text{center}} \leq x_{\text{low},_o} - (\rho + \alpha_{N-1}) + M b^p_{\text{circ-obst},_{o1}} \tag{46a}$$

$$y^p_{\text{center}} \leq y_{\text{low},_o} - (\rho + \alpha_{N-1}) + M b^p_{\text{circ-obst},_{o2}} \tag{46b}$$

$$x^p_{\text{center}} \geq x_{\text{high},_o} + (\rho + \alpha_{N-1}) - M b^p_{\text{circ-obst},_{o3}} \tag{46c}$$

$$y^p_{\text{center}} \geq y_{\text{high},_o} + (\rho + \alpha_{N-1}) - M b^p_{\text{circ-obst},_{o4}} \tag{46d}$$

$$\sum_{i=1}^{4} b^p_{\text{circ-obst},_{oi}} \leq 3. \tag{46e}$$

4) Interconnected Constraints (13): Over the planning horizon, the coupling constraints include vehicle avoidance constraints

$$x^p_{k+j\,|\,k} \leq x^q_{k+j\,|\,k} - d^{pq}_{\text{total}} + M b^{pq}_{\text{veh},_{j1}} \tag{47a}$$

$$y^p_{k+j\,|\,k} \leq y^q_{k+j\,|\,k} - d^{pq}_{\text{total}} + M b^{pq}_{\text{veh},_{j2}} \tag{47b}$$

$$x^p_{k+j\,|\,k} \geq x^q_{k+j\,|\,k} + d^{pq}_{\text{total}} - M b^{pq}_{\text{veh},_{j3}} \tag{47c}$$

$$y^p_{k+j\,|\,k} \geq y^q_{k+j\,|\,k} + d^{pq}_{\text{total}} - M b^{pq}_{\text{veh},_{j4}} \tag{47d}$$

$$\sum_{i=1}^{4} b^{pq}_{\text{veh},_{ji}} \leq 3 \tag{47e}$$

where

$$d^{pq}_{\text{total}} = \begin{cases} 2d + 2\alpha_j, & q < p \\ 2d + \alpha_j + \alpha_{j+1}, & q > p. \end{cases}$$

Beyond the planning horizon, constraints on the safety circles ensure the vehicle avoidance

$$x^p_{\mathrm{center}} \le x^q_{\mathrm{center}} - 2(\rho + d + \alpha_{N-1}) + M b^{pq}_{\mathrm{circ},_1} \tag{48a}$$
$$y^p_{\mathrm{center}} \le y^q_{\mathrm{center}} - 2(\rho + d + \alpha_{N-1}) + M b^{pq}_{\mathrm{circ},_2} \tag{48b}$$
$$x^p_{\mathrm{center}} \ge x^q_{\mathrm{center}} + 2(\rho + d + \alpha_{N-1}) - M b^{pq}_{\mathrm{circ},_3} \tag{48c}$$
$$y^p_{\mathrm{center}} \ge y^q_{\mathrm{center}} + 2(\rho + d + \alpha_{N-1}) - M b^{pq}_{\mathrm{circ},_4} \tag{48d}$$
$$\sum_{i=1}^{4} b^{pq}_{\mathrm{circ},_i} \le 3. \tag{48e}$$

5) Objective Function (20): The objective function uses a binary variable $\boldsymbol{b}^p_{\mathrm{vis}}$ to select one visible point $\boldsymbol{r}^p_{\mathrm{vis}}$ from a list of cost points, from which the cost-to-go is known. Let $\boldsymbol{r}_{\mathrm{cp},_i}$ denote the ith cost point and $i = 1, \ldots, n_{\mathrm{cp}}$, where n_{cp} is a number of cost points. Then

$$\boldsymbol{r}^p_{\mathrm{vis}} = \sum_{i=1}^{n_{\mathrm{cp}}} b^p_{\mathrm{vis},_i} \boldsymbol{r}^p_{\mathrm{cp},_i} \tag{49a}$$
$$\sum_{i=1}^{n_{\mathrm{cp}}} b^p_{\mathrm{vis},_i} = 1 \tag{49b}$$
$$\tilde{f}^p\left(\boldsymbol{r}^p_{\mathrm{vis}}\right) = \sum_{i=1}^{n_{\mathrm{cp}}} b^p_{\mathrm{vis},_i} \tilde{f}^p\left(\boldsymbol{r}^p_{\mathrm{cp},_i}\right) \tag{49c}$$
$$J^p \ge [\cos\theta_m, \sin\theta_m]\left(\boldsymbol{r}^p_{k+N\,|\,k} - \boldsymbol{r}^p_{\mathrm{vis}}\right) + \tilde{f}^p\left(\boldsymbol{r}^p_{\mathrm{vis}}\right), \ \forall m \tag{49d}$$

where the cost-to-go $\tilde{f}^p(\boldsymbol{r}^p_{\mathrm{cp},_i})$ from each cost point to the target of vehicle p is calculated prior to MILP and is constant in MILP. To ensure the visibility of the selected cost point $\boldsymbol{r}^p_{\mathrm{vis}}$ from the terminal point $\boldsymbol{x}^p_{k+N\,|\,k}$, obstacle avoidance constraints are enforced on n_{int} interpolation points that are placed on the line connecting $\boldsymbol{r}^p_{\mathrm{vis}}$ and $\boldsymbol{x}^p_{k+N\,|\,k}$

$$\mu_l x^p_{k+N\,|\,k} + (1-\mu_l)x^p_{\mathrm{vis}} \le x_{\mathrm{low},_o} - \alpha_{N-1} + M b^p_{\mathrm{int},_{lo1}} \tag{50a}$$
$$\mu_l y^p_{k+N\,|\,k} + (1-\mu_l)y^p_{\mathrm{vis}} \le y_{\mathrm{low},_o} - \alpha_{N-1} + M b^p_{\mathrm{int},_{lo2}} \tag{50b}$$
$$\mu_l x^p_{k+N\,|\,k} + (1-\mu_l)x^p_{\mathrm{vis}} \ge x_{\mathrm{high},_o} + \alpha_{N-1} - M b^p_{\mathrm{int},_{lo3}} \tag{50c}$$
$$\mu_l y^p_{k+N\,|\,k} + (1-\mu_l)y^p_{\mathrm{vis}} \ge y_{\mathrm{high},_o} + \alpha_{N-1} - M b^p_{\mathrm{int},_{lo4}} \tag{50d}$$
$$\sum_{i=1}^{4} b^p_{\mathrm{int},_{loi}} \le 3 \tag{50e}$$
$$\mu_l = \frac{l}{n_{\mathrm{int}}}, \quad l = 1, \ldots, n_{\mathrm{int}}.$$

In summary, the MILP implementation of subproblem $\mathsf{P}^p(\boldsymbol{x}^p_k)$ is to minimize J^p in (49d) subject to (9), (10), and (41a)–(50e). The optimization variables are $\boldsymbol{r}^p_{k+j\,|\,k}, \boldsymbol{v}^p_{k+j\,|\,k}, \boldsymbol{a}^p_{k+j\,|\,k}, \boldsymbol{O}^p_{\mathrm{L}}, \boldsymbol{O}^p_{\mathrm{R}}, \boldsymbol{O}^p, \boldsymbol{r}^p_{\mathrm{vis}}$, and all binary variables b^p's.

References

[1] M. B. Milam, K. Mushambi, and R. M. Murray, "A new computational approach to real-time trajectory generation for constrained mechanical systems," in *Proc. IEEE Conf. Dec. Control*, 2000, pp. 845–851.

[2] J. Bellingham, A. Richards, and J. How, "Receding horizon control of autonomous aerial vehicles," in *Proc. IEEE Amer. Control Conf.*, 2002, pp. 3741–3746.

[3] W. B. Dunbar and R. Murray, "Model predictive control of coordinated multi-vehicle formations," in *Proc. IEEE Conf. Dec. Control*, 2002, pp. 4631–4636.

[4] D. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *Proc. IEEE Conf. Dec. Control*, 2003, pp. 3621–3626.

[5] J. A. Primbs, "The analysis of optimization based controllers," *Automatica*, vol. 37, no. 6, pp. 933–938, 2001.

[6] J. M. Maciejowski, *Predictive Control with Constraints*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[7] J. A. Rossiter, *Model-Based Predictive Control: A Practical Approach*. Boca Raton, FL: CRC, 2003.

[8] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[9] P. O. M. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Autom. Control*, vol. 8, no. 43, pp. 1136–1142, Aug. 1998.

[10] A. Casavola, M. Giannelli, and E. Mosca, "Min-max predictive control strategies for input-saturated polytopic uncertain systems," *Automatica*, vol. 36, pp. 125–133, 2000.

[11] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.

[12] L. Chisci, J. A. Rossiter, and G. Zappa, "Systems with persistent disturbances: Predictive control with restricted constraints," *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.

[13] E. C. Kerrigan and J. M. Maciejowski, "Robust feasibility in model predictive control," in *Proc. IEEE Conf. Dec. Control*, 2001, pp. 728–733.

[14] F. A. Cuzzolaa, J. C. Geromel, and M. Morari, "An improved approach for constrained robust model predictive control," *Automatica*, vol. 38, no. 7, pp. 1183–1189, 2002.

[15] J. Lofberg, "Minimax approaches to robust model predictive control," Ph.D. dissertation, Dept. Electr. Eng., Linkoping Univ., Linkoping, Sweden, 2003.

[16] A. Richards, "Robust constrained model predictive control," Ph.D. dissertation, Dept. Aeronautics Astronautics, MIT, Cambridge, MA, 2004.

[17] A. Richards and J. How, "A Decentralized algorithm for robust constrained model predictive control," in *Proc. IEEE Amer. Control Conf.*, Boston, MA, 2004, pp. 4261–4266.

[18] A. Richards and J. How, "Decentralized model predictive control of cooperating UAVs," in *Proc. IEEE Conf. Dec. Control*, 2004, pp. 4286–4291.

[19] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Syst. Mag.*, vol. 2, no. 1, pp. 44–52, Feb. 2002.

[20] D. Jia and B. Krogh, "Min-max feedback model predictive control for distributed control with communication," in *Proc. IEEE Amer. Control Conf.*, 2002, pp. 4507–45.

[21] S. L. Waslander, G. Inalhan, and C. J. Tomlin, "Decentralized optimization via Nash bargaining," in *Theory and Algorithms for Cooperative Systems*. Singapore: World Scientific, 2004.

[22] T. Keviczky, F. Borrelli, and G. Balas, "A study on decentralized receding horizon control for decoupled systems," in *Proc. IEEE Amer. Control Conf.*, 2004, pp. 4921–4926.

[23] T. Schouwenaars, J. How, and E. Feron, "Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees," in *Proc. AIAA Guid., Nav., Control Conf.*, 2004, pp. 6820–6825.

[24] M. Flint, M. Polycarpou, and E. Fernandez-Gaucherand, "Cooperative path-planning for autonomous vehicles using dynamics programming," in *Proc. 15th IFAC World Congress*, 2002, pp. 1694–1699.

[25] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Plant-wide optimal control with decentralized MPC," presented at the Int. Symp. Synamics Control Process Syst., Boston, MA, 2004.

[26] Y. Kuwata, A. Richards, T. Schouwenaars, and J. How, "Robust constrained receding horizon control for trajectory planning," presented at the AIAA Guid., Nav., Control Conf., 2005, AIAA 2005-6079.

[27] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proc. IEEE Amer. Control Conf.*, 2004, pp. 5576–5581.

[28] I. Kolmanovsky and E. G. Gilbert, "Maximal output admissible sets for discrete-time systems with disturbance inputs," in *Proc. IEEE Amer. Control Conf.*, 1995, pp. 1995–1999.

[29] E. C. Kerrigan, "Robust constraint satisfaction invariant sets and predictive control," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., Nov. 2000.

[30] C. Tin, "Robust multi-UAV planning in dynamic and uncertain environments," Master's thesis, Dept. Mechan. Eng., MIT, Cambridge, MA, 2004.

[31] Y. Kuwata and J. How, "Three dimensional receding horizon control for UAVs," presented at the AIAA Guid., Nav., Control Conf., 2004, AIAA 2004-5144.

[32] Y. Kuwata and J. How, "Stable trajectory design for highly constrained environments using receding horizon control," in *Proc. IEEE Amer. Control Conf.*, 2004, pp. 902–907.

[33] S. Skiena, *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica.* Boston, MA: Addison-Wesley, 1990.

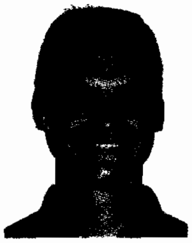
[34] S. Park, J. Deyst, and J. P. How, "A new nonlinear guidance logic for trajectory tracking," presented at the AIAA Guid., Nav., Control Conf., 2004, AIAA 2004-4900.

Yoshiaki Kuwata (S'06) received the B.Eng. degree from the University of Tokyo, Tokyo, Japan, in 2001, and the S.M. and Ph.D. degrees in aeronautics and astronautics from Massachusetts Institute of Technology (MIT), Cambridge, in 2003 and 2007, respectively.

He is currently a Postdoctoral Associate at MIT working for the DARPA Grand Challenge. His research interests include cooperative control, receding horizon control, and path planning for unmanned vehicles.

Arthur Richards (M'05) received the Masters of Engineering degree from the University of Cambridge, Cambridge, U.K., and the Masters of Science and Ph.D. degrees in aeronautics and astronautics from the Massachusetts Institute of Technology, Cambridge.

He became a lecturer at the University of Bristol, Bristol, U.K., in 2004. His current research interests include optimization, control, and autonomy for aerospace systems.



Tom Schouwenaars received the Ph.D. degree in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT), Cambridge, in 2006, and the M.Sc. degree in electrical engineering from the Catholic University of Leuven, Leuven, Belgium, in 2001.

Currently, he is an Associate in the Quantitative Strategies Division, Goldman, Sachs and Co., NY. At MIT, he worked in the Laboratory for Information and Decision Systems under the supervision of Prof. E. Feron and Prof. J. How. His research was on safe trajectory planning for unmanned vehicles using mixed integer linear programming, which was successfully demonstrated on an autonomous Boeing aircraft during Darpa's Software Enabled Control Program.

Dr. Schouwenaars is a Francqui Fellow of the Belgian American Educational Foundation and was awarded the 2005 AIAA Unmanned Aerial Vehicles Graduate Award.

Jonathan P. How (SM'05) received the B.A.Sc. degree from the University of Toronto, Toronto, ON, Canada, in 1987, and the S.M. and Ph.D. degrees in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT), Cambridge, in 1990 and 1993, respectively.

He is an Associate Professor in the Department of Aeronautics and Astronautics, MIT. After receiving the Ph.D. degree, he then studied for two years at MIT as a postdoctoral associate for the Middeck Active Control Experiment (MACE) that flew on-board the Space Shuttle Endeavour in March 1995. Prior to joining MIT in 2000, he was an Assistant Professor in the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA. His current research interests include using operations research tools such as mixed-integer programming to optimize the coordination and control of autonomous vehicles in dynamic uncertain environments.

Dr. How was a recipient of the 2002 Institute of Navigation Burka Award. He is the Raymond L. Bisplinghoff Fellow for the MIT Aero/Astro Department and an Associate Fellow of AIAA.

International Journal of Control
Vol. 80, No. 9, September 2007, 1517–1531

Taylor & Francis
Taylor & Francis Group

Robust distributed model predictive control

A. RICHARDS*† and J. P. HOW‡

†Lecturer, Department of Aerospace Engineering, University of Bristol,
Queens Building, University Walk, Bristol, BS8 1TR, UK
‡Associate Professor, Massachusetts Institute of Technology,
77 Massachusetts Avenue, Cambridge, MA 02139, USA

(*Received 25 July 2006; in final form 4 June 2007*)

This paper presents a formulation for distributed model predictive control (DMPC) of systems with coupled constraints. The approach divides the single large planning optimization into smaller sub-problems, each planning only for the controls of a particular subsystem. Relevant plan data is communicated between sub-problems to ensure that all decisions satisfy the coupled constraints. The new algorithm guarantees that all optimizations remain feasible, that the coupled constraints will be satisfied, and that each subsystem will converge to its target, despite the action of unknown but bounded disturbances. Simulation results are presented showing that the new algorithm offers significant reductions in computation time for only a small degradation in performance in comparison with centralized MPC.

1. Introduction

This paper presents a distributed form of model predictive control (MPC) for problems involving multiple subsystems, each with independent dynamics, but subjected to coupled constraints. Uncertainty is considered in the form of persistent, independent, affine disturbances acting upon each subsystem. For each subsystem, a local planning optimization is solved once per timestep, without iteration, giving a more scalable computation than a single large optimization for the whole system. The subsystems communicate with each other, enabling the coupling between them to be enforced in each optimization. The algorithm guarantees robust satisfaction of the coupled constraints, feasibility of all optimizations involved, and convergence to specified targets.

Centralized MPC, using numerical optimization for online replanning, has been widely developed and applied for constrained systems (Qin and Badgwell 1997, Maciejowski 2002), with many results concerning stability (Bemporad and Morari 1999, Mayne *et al.* 2000) and robustness (Scokaert and Mayne 1998, Kerrigan and Maciejowski 2001, Richards and How 2006). However, the computational effort required for the optimization scales poorly with the size of the system and can become prohibitive for very large systems. To address this computational issue, attention has recently focused on distributed MPC (DMPC) (Camponogara *et al.* 2002), breaking the optimization into smaller sub-problems, with the rationale that solving many small problems is faster and more scalable than solving one large problem. Furthermore, in applications such as the control of multiple vehicles (Pachter and Chandler 1998), large chemical plants (Venkat *et al.* 2004) or communication networks (Yan and Bitmead 2005), there is a spatial separation of control agents, and it is therefore natural to distribute the control computation. The challenge of this approach is to ensure that the distributed decision making leads to actions that are consistent with the actions of others and satisfy the coupling constraints. Various approaches have been investigated, including treating the influence of other subsystems as an unknown disturbance (Jia and Krogh 2002, Magni and Scatolini 2006), coupling penalty functions (Shim *et al.* 2003, Waslander *et al.* 2003,

*Corresponding author. Email: arthur.richards@bristol.ac.uk

International Journal of Control
ISSN 0020-7179 print ISSN 1366-5820 online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207170701491070

Dunbar and Murray 2006), partial grouping of computations (Keviczky *et al.* 2006), loitering options for safety guarantees (Schouwenaars *et al.* 2004) and dynamic programming (Flint *et al.* 2002). Some approaches involve iterative negotiations between subsystems (Waslander *et al.* 2003, Venkat *et al.* 2004) and apply game theory or duality (Raffard *et al.* 2004) to study convergence. Ling *et al.* (2005) propose a sequential solution scheme with similar features to the new method in this paper, but applied to a system with coupled dynamics and no uncertainty. Distributed MPC is also related to distributed optimization (Singh and Titli 1978), with the distinction that the latter is primarily concerned with finding optimal or near-optimal solutions, whereas the former commonly sacrifices optimality in favour of fast computation. Distribution is further complicated when disturbances act on the subsystems, making the prediction of future behaviour uncertain. The new DMPC algorithm in this paper is unique, to the authors' knowledge, in addressing the issue of feasibility and convergence for systems with coupling constraints and persistent disturbances.

The key features of the new DMPC algorithm are that each subsystem (e.g., a vehicle) only solves a sub-problem for its own plan, and each of these sub-problems is solved only once per time step, without iteration. The method employs at each time step a sequential solution procedure, outlined in figure 1(a). Under the assumption of bounded disturbances, each sub-problem is guaranteed to be feasible, thus ensuring robust constraint satisfaction across the group, and its cost is proven to decrease, implying that the system converges to the target. The plan data relevant to the coupled constraints is then communicated among the subsystems. Figure 1(b) shows the information requirements for sub-problem p. Each sub-problem accommodates (i) the latest plans of those subsystems earlier in the sequence and (ii) predicted plans of those later in the sequence. The principle of constraint tightening, used for centralized MPC in Chisci *et al.* (2001) and Richards and How (2006) to accommodate uncertainty in future behaviour, is applied in the new algorithm to account for uncertainty in the actions of other subsystems. A key advantage of the constraint tightening approach is that the complexity of the optimization remains the same as for the nominal problem, which is important for the scalability of the new algorithm. At initialization, in common with most constrained MPC formulations, e.g., Mayne *et al.* (2000), Chisci *et al.* (2001) and Ling *et al.* (2005), it is necessary to find a feasible solution to the centralized problem, but this need not be optimal.

By adopting a sequential updating strategy, as outlined in figure 1(a), in which only one subsystem optimizes at a time and the results are passed on to its neighbours, we are able to achieve firm guarantees of feasibility and convergence for the class of problems considered. The improved scalability follows because the time to solve an optimization grows at least with the cube of the number of inputs (Rao *et al.* 1998) or worse in more complex cases (Richards and How 2004). Therefore, if we consider an example problem of n subsystems each with m inputs, giving a total of mn inputs to the system, the centralized solution time grows as $O((mn)^3)$. However, the distributed solution time, allowing for the sequential process, grows only as $O(m^3n)$ with a reduced dependency on n. Additional benefits accrue because the computation delay in each local loop is equal only to the time to solve the local sub-problem, not the whole sequence (Richards and How 2005). Furthermore, parallel computation can be exploited in typical cases (Kuwata *et al.* 2006), and we show how structure in the constraints makes this possible.

Section 2 presents the problem statement for DMPC in a general form. Section 3 presents centralized MPC solution for this problem, used for comparison and initialization later in the paper. Section 4 develops the DMPC algorithm and § 5 proves its feasibility and convergence properties. Section 6 investigates the communication requirements for DMPC: while the method relies on the ability to communicate intentions between subsystems, it is possible to make use of structure in the constraints such that only relevant data is exchanged. Finally, § 7 presents illustrative examples of DMPC in simulation, comparing performance and solution times with centralized MPC.

2. Problem statement

Consider the problem of controlling N_p subsystems. Each subsystem, denoted by subscript $p \in \{1, \dots, N_p\}$, has linear, time-invariant, discretized dynamics

$$\mathbf{x}_p(k+1) = \mathbf{A}_p\mathbf{x}_p(k) + \mathbf{B}_p\mathbf{u}_p(k) + \mathbf{w}_p(k), \qquad (1)$$

where $\mathbf{x}_p(k) \in \Re^{N_p^x}$ is the state vector of subsystem p at time k, $\mathbf{u}_p(k) \in \Re^{N_p^u}$ is the control input to subsystem p and $\mathbf{w}_p(k) \in \Re^{N_p^x}$ is the disturbance acting upon subsystem p. Assume all subsystems $(\mathbf{A}_p, \mathbf{B}_p)$ are controllable and the complete states $\mathbf{x}_p$ are available. The disturbances are unknown *a priori* but lie in known independent bounded sets

$$\forall k, p \quad \mathbf{w}_p(k) \in \mathcal{W}_p \subset \Re^{N_p^x}. \qquad (2)$$

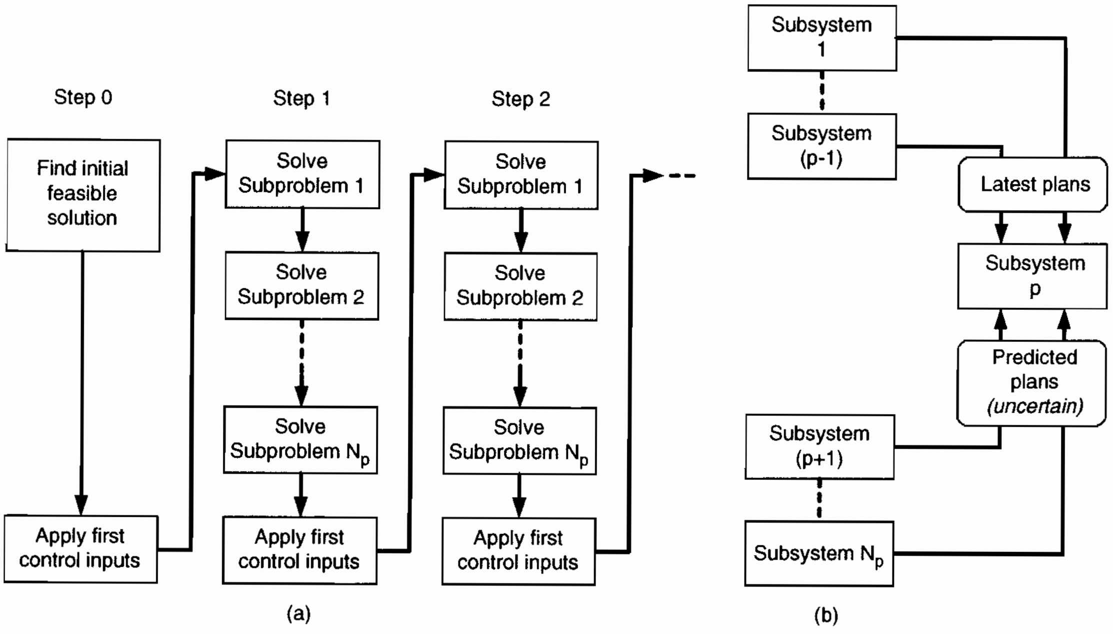


Figure 1. Overview of distributed algorithm. (a) Procedure flow; (b) Information flow.

Each subsystem is subjected to the following local constraints upon outputs $\mathbf{y}_p(k) \in \Re^{N_p^y}$

$$\mathbf{y}_p(k) = \mathbf{C}_p\mathbf{x}_p(k) + \mathbf{D}_p\mathbf{u}_p(k) \tag{3}$$

$$\mathbf{y}_p(k) \in \mathcal{Y}_p \subset \Re^{N_p^y}\ \forall k, \tag{4}$$

where the matrices $\mathbf{C}_p$ and $\mathbf{D}_p$ and the sets $\mathcal{Y}_p$ for each p are all chosen by the designer.

The whole system is subjected to a set of N_c coupling constraints applied to outputs $\mathbf{z}_{cp}(k) \in \Re^{N_c^z}$ summed across the subsystems,

$$\forall c \in \{1, \dots, N_c\}:$$

$$\mathbf{z}_{cp}(k) = \mathbf{E}_{cp}\mathbf{x}_p(k) + \mathbf{F}_{cp}\mathbf{u}_p(k) \quad \forall p \in \{1, \dots, N_p\} \tag{5}$$

$$\sum_{p=1}^{N_p} \mathbf{z}_{cp}(k) \in \mathcal{Z}_c \subset \Re^{N_c^z}\ \forall k, \tag{6}$$

where the matrices $\mathbf{E}_{cp}$ and $\mathbf{F}_{cp}$ for each c and p and the sets $\mathcal{Z}_c$ for each c are all chosen by the designer. The division of the coupling into the N_c separate constraints will be exploited in §6 to identify communication requirements based on structure.

The objective is for tracking outputs $\mathbf{s}_p(k) \in \Re^{N_p^s}$ for each subsystem to converge to target sets

$$\mathbf{s}_p(k) = \mathbf{G}_p\mathbf{x}_p(k) + \mathbf{H}_p\mathbf{u}_p(k) \tag{7}$$

$$\mathbf{s}_p(k) \to \mathcal{S}_p \quad \text{as } k \to \infty, \tag{8}$$

where the matrices $\mathbf{G}_p$ and $\mathbf{H}_p$ and the compact sets $\mathcal{S}_p$ for each p are all chosen by the designer. Note that this represents a decoupled objective: coupling between subsystems is only via the constraints on outputs $\mathbf{z}_{cp}(k)$ in this problem statement. This captures an important class of problems, including, for example, maneuvering a group of vehicles from one waypoint to the next while maintaining relative formation and/or avoiding collisions.

3. Review of centralized robust MPC

This section presents a short review of the centralized form of the robust MPC problem for the problem in §7. It is identical to the controller in Richards and How (2006), treating the group of subsystems as a single system and solving a single optimization for the controls. The centralized formulation is relevant here for three reasons: first, because a feasible solution to the centralized problem is required to intialize the distributed algorithm; second, because demonstrating satisfaction of the centralized constraints is an intermediate step in proving robustness of the distributed algorithm; and third, because centralized MPC is used in the examples in §7 for comparison with the new distributed controller.

The online optimization approximates the complete problem in §2 by solving it over a finite horizon of N steps. A control-invariant terminal set constraint is applied to ensure stability (Mayne *et al.* 2000). Predictions are made using the nominal system model, i.e., (1) without the disturbance term. The output constraints are tightened in a monotonic sequence (Richards and How 2006) to ensure robust feasibility, retaining a margin based on a particular candidate policy.

Define the centralized optimization problem $\bar{P}(\bar{\mathbf{x}}(k))$

$$J^*_C(\bar{\mathbf{x}}(k)) = \min_{\bar{\mathbf{U}}(k)} J_C\big(\bar{\mathbf{x}}(k), \bar{\mathbf{U}}(k)\big)$$

$$= \min_{\bar{\mathbf{U}}(k)} \sum_{p=1}^{N_p} \sum_{j=0}^{N-1} d\big(\mathbf{s}_p(k+j|k), \mathcal{S}_p(j)\big)$$

subject to $\forall j \in \{0, \dots, N-1\} \quad \forall p \in \{1, \dots, N_p\}$

$$\mathbf{x}_p(k|k) = \mathbf{x}_p(k) \tag{9a}$$

$$\mathbf{x}_p(k+j+1|k) = \mathbf{A}_p\mathbf{x}_p(k+j|k) + \mathbf{B}_p\mathbf{u}_p(k+j|k) \tag{9b}$$

$$\mathbf{y}_p(k+j|k) = \mathbf{C}_p\mathbf{x}_p(k+j|k) + \mathbf{D}_p\mathbf{u}_p(k+j|k) \tag{9c}$$

$$\mathbf{z}_{cp}(k+j|k) = \mathbf{E}_{cp}\mathbf{x}_p(k+j|k) + \mathbf{F}_{cp}\mathbf{u}_p(k+j|k) \quad \forall c \in \{1, \dots, N_c\} \tag{9d}$$

$$\mathbf{s}_p(k+j|k) = \mathbf{G}_p\mathbf{x}_p(k+j|k) + \mathbf{H}_p\mathbf{u}_p(k+j|k) \tag{9e}$$

$$\mathbf{x}_p(k+N|k) \in \mathcal{T}_p \tag{9f}$$

$$\mathbf{y}_p(k+j|k) \in \mathcal{Y}_p(j) \tag{9g}$$

$$\sum_{p=1}^{N_p} \mathbf{z}_{cp}(k+j|k) \in \bar{\mathcal{Z}}_c(j) \quad \forall c \in \{1, \dots, N_c\}, \tag{9h}$$

where

- the double index notation $(k+j|k)$ denotes a prediction for j steps ahead from time k
- the state $\bar{\mathbf{x}}(k)$ denotes the combined state of all subsystems $\{\mathbf{x}_1(k), \dots, \mathbf{x}_{N_p}(k)\}$
- the decision variable $\bar{\mathbf{U}}(k)$ denotes combined control sequences for all subsystems $\{\mathbf{U}_1(k), \dots, \mathbf{U}_{N_p}(k)\}$ where each $\mathbf{U}_p(k)$ denotes a sequence $\{\mathbf{u}_p(k|k), \dots, \mathbf{u}_p(k+N-1|k)\}$
- $d(\mathbf{s}_p, \mathcal{S}_p)$ denotes a distance metric (Kerrigan and Maciejowski 2004)

$$d(\mathbf{s}_p, \mathcal{S}_p) = \min_{\hat{\mathbf{s}}_p \in \mathcal{S}_p} \|\mathbf{s_p} - \hat{\mathbf{s}}_p\| \tag{10}$$

- the local constraint sets $\mathcal{Y}_p(j)$ are tightened at future plan steps in a monotonic sequence, ensuring the existence of a margin to allow for future feedback action in response to disturbances (Chisci *et al.* 2001, Richards and How 2006). The tightening is embodied in the following recursions

$$\forall p \in \{1, \dots, N_p\} : \mathcal{Y}_p(0) = \mathcal{Y} \tag{11a}$$

$$\mathcal{Y}_p(j+1) = \mathcal{Y}_p(j) \sim \big(\mathbf{C}_p + \mathbf{D}_p\mathbf{K}_p(j)\big)\mathbf{L}_p(j)\mathcal{W}_p, \quad \forall j \in \{0, \dots, N-1\} \tag{11b}$$

- the tightened target sets $\mathcal{S}_p(j)$ are found using the following similar recursions (Richards and How 2006)

$$\forall p \in \{1, \dots, N_p\} \ : \mathcal{S}_p(0) = \mathcal{S} \tag{12a}$$

$$\mathcal{S}_p(j+1) = \mathcal{S}_p(j) \sim \big(\mathbf{G}_p + \mathbf{H}_p\mathbf{K}_p(j)\big)\mathbf{L}_p(j)\mathcal{W}_p, \quad \forall j \in \{0, \dots, N-1\} \tag{12b}$$

- the tightened coupling constraint sets $\bar{\mathcal{Z}}_c(j)$ are found using the following recursions

$$\forall c \in \{1, \dots, N_c\} \ : \bar{\mathcal{Z}}_c(0) = \mathcal{Z}_c \tag{13a}$$

$$\bar{\mathcal{Z}}_c(j+1) = \bar{\mathcal{Z}}_c(j) \sim [(\mathbf{E}_{c1} + \mathbf{F}_{c1}\mathbf{K}_1(j))\mathbf{L}_1(j)\mathcal{W}_1 \oplus \cdots \oplus \big(\mathbf{E}_{cN_p} + \mathbf{F}_{cN_p}\mathbf{K}_{N_p}(j)\big)\mathbf{L}_{N_p}(j)\mathcal{W}_{N_p}\big], \quad \forall j \in \{0, \dots, N-1\} \tag{13b}$$

- the operator "$\sim$" represents the Pontryagin difference (Kolmanovsky and Gilbert 1998)

$$\mathcal{A} \sim \mathcal{B} = \{\mathbf{a} \mid \mathbf{a} + \mathbf{b} \in \mathcal{A}, \ \forall \mathbf{b} \in \mathcal{B}\} \tag{14}$$

which has the following useful property, to be used in § 5 to prove the properties of the DMPC algorithm

$$\mathbf{c} \in (\mathcal{A} \sim \mathcal{B}) \Rightarrow \mathbf{c} + \mathbf{b} \in \mathcal{A} \quad \forall \mathbf{b} \in \mathcal{B} \tag{15}$$

- the matrices $\mathbf{K}_p(j)\, j \in \{0, \dots, N-1\}$ in (11), (12) and (13) are candidate control laws for each subsystem, chosen by the designer. A constant gain may be used if required, but Richards and How (2006) discuss the benefits of the more general time-varying gain. The associated state transition matrices $\mathbf{L}_p(j)$ are given by

$$\mathbf{L}_p(0) = \mathbf{I} \tag{16a}$$

$$\mathbf{L}_p(j+1) = \big(\mathbf{A}_p + \mathbf{B}_p\mathbf{K}_p(j)\big)\mathbf{L}_p(j), \quad \forall j \in \{0, \dots N-2\} \tag{16b}$$

- the terminal constraint sets $\mathcal{T}_p$ in (9f) are robustly positive invariant sets (Kolmanovsky and Gilbert 1998), satisfying

$$\big(\mathbf{A}_p + \mathbf{B}_p\mathbf{K}_p(N-1)\big)\mathbf{x}_p + \big(\mathbf{A}_p + \mathbf{B}_p\mathbf{K}_p(N-1)\big)\mathbf{L}_p(N-1)\mathbf{w}_p \in \mathcal{T}_p, \quad \forall \mathbf{x}_p \in \mathcal{T}_p, \quad \forall \mathbf{w}_p \in \mathcal{W}_p, \ \forall p \in \{1, \dots, N_p\} \tag{17a}$$

$$\big(\mathbf{C}_p\mathbf{x}_p + \mathbf{D}_p\mathbf{K}_p(N-1)\big)\mathbf{x}_p \in \mathcal{Y}_p(N), \quad \forall \mathbf{x}_p \in \mathcal{T}_p, \quad \forall p \in \{1, \dots, N_p\} \tag{17b}$$

$$\sum_{p=1}^{N_p} \big(\mathbf{E}_{cp} + \mathbf{F}_{cp}\mathbf{K}_p(N-1)\big)\mathbf{x}_p \in \bar{\mathcal{Z}}_c(N), \quad \forall c \in \{1, \dots, N_c\}, \quad \forall (\mathbf{x}_1^T, \dots, \mathbf{x}_{N_p}^T)^T \in \{\mathcal{T}_1 \times \cdots \times \mathcal{T}_{N_p}\} \tag{17c}$$

$$\big(\mathbf{G}_p\mathbf{x}_p + \mathbf{H}_p\mathbf{K}_p(N-1)\big)\mathbf{x}_p \in \mathcal{S}_p(N), \quad \forall \mathbf{x}_p \in \mathcal{T}_p, \ \forall p \in \{1, \dots, N_p\}. \tag{17d}$$

Tools for determining suitable sets are available (Kerrigan 2005). Note that if the candidate controllers $\mathbf{K}_p(\cdot)$ are chosen such that they would drive the undisturbed system to the origin in N steps or fewer, then $\mathbf{L}_p(N-1)=\mathbf{0}$ and the requirement of robust control invariance in (17a) is relaxed to nominal invariance, and simple forms such as equilibrium points can be employed as terminal constraints (Richards and How 2006). The centralized optimization is employed in the following algorithm.

Algorithm 1: *Centralized MPC.*

(i) *Set* $k=0$.
(ii) *Find a solution to the centralized optimization problem* $\bar{P}(\bar{\mathbf{x}}(k))$. *If a solution cannot be found, stop.*
(iii) *Apply control* $\mathbf{u}_p^*(k|k)$ *to each subsystem p.*
(iv) *Increment k and go to Step* (ii).

Given initial feasibility of $\bar{P}(\bar{\mathbf{x}}(0))$, this controller guarantees feasibility of all optimizations $\bar{P}(\bar{\mathbf{x}}(k))$, satisfaction of constraints (4) and (6), and convergence to the targets (8), according to Richards and How (2006).

4. Distributed MPC algorithm

This section describes a distributed algorithm for solving the problem in §2. The centralized optimization $\bar{P}$ from §3 is divided into a sequence of N_p sub-problems, shown in figure 1, with each sub-problem solving for the trajectory of only one subsystem.

Each sub-problem p solves for future controls of subsystem p only. The current state $\mathbf{x}_p(k)$ of subsystem p and the output information for other subsystems $\tilde{\mathbf{Z}}_p(k)$ are constant parameters, obtained by measurement and communication, respectively. It is this communication of coupling information from other subsystems that ensures consistency across the whole system. The sub-problem $P_p\left(\mathbf{x}_p(k),\tilde{\mathbf{Z}}_p(k)\right)$ is defined as follows:

$$J_p^*\left(\mathbf{x}_p(k),\tilde{\mathbf{Z}}_p(k)\right)=\min_{\mathbf{U}_p(k)} J_p\left(\mathbf{x}_p(k),\tilde{\mathbf{Z}}_p(k),\mathbf{U}_p(k)\right)$$

$$=\min_{\mathbf{U}_p(k)}\sum_{j=0}^{N-1} d\left(\mathbf{s}_p(k+j|k),\mathcal{S}_p(j)\right)$$

$$\text{subject to } \forall j\in\{0,\dots,N-1\}$$

$$\mathbf{x}_p(k|k)=\mathbf{x}_p(k) \tag{18a}$$

$$\mathbf{x}_p(k+j+1|k)=\mathbf{A}_p\mathbf{x}_p(k+j|k)+\mathbf{B}_p\mathbf{u}_p(k+j|k) \tag{18b}$$

$$\mathbf{y}_p(k+j|k)=\mathbf{C}_p\mathbf{x}_p(k+j|k)+\mathbf{D}_p\mathbf{u}_p(k+j|k) \tag{18c}$$

$$\mathbf{z}_{cp}(k+j|k)=\mathbf{E}_{cp}\mathbf{x}_p(k+j|k)+\mathbf{F}_{cp}\mathbf{u}_p(k+j|k)\ \ \forall c\in\{1,\dots,N_c\} \tag{18d}$$

$$\mathbf{s}_p(k+j|k)=\mathbf{G}_p\mathbf{x}_p(k+j|k)+\mathbf{H}_p\mathbf{u}_p(k+j|k) \tag{18e}$$

$$\mathbf{x}_p(k+N|k)\in\mathcal{T}_p \tag{18f}$$

$$\mathbf{y}_p(k+j|k)\in\mathcal{Y}_p(j) \tag{18g}$$

$$\mathbf{z}_{cp}(k+j|k)+\tilde{\mathbf{z}}_{cp}(k+j|k)\in\mathcal{Z}_{cp}(j)\ \ \forall c\in\{1,\dots,N_c\} \tag{18h}$$

where

- the communicated coupling outputs $\tilde{\mathbf{Z}}_p(k)$ consist of output sequences for each constraint $\{\tilde{\mathbf{Z}}_{1p}(k),\dots,\tilde{\mathbf{Z}}_{N_cp}(k)\}$ in which each element $\tilde{\mathbf{Z}}_{cp}(k)$ is a sequence of output values $\{\tilde{\mathbf{z}}_{cp}(k|k),\dots,\tilde{\mathbf{z}}_{cp}(k+N-1|k)\}$. Each communicated output value $\tilde{\mathbf{z}}_{cp}(k+j|k)$ has two components, as shown in figure 1(b): (i) the most recent plans of those subsystems earlier than p in the planning sequence and (ii) predicted plans for subsystems later in the sequence

$$\tilde{\mathbf{z}}_{cp}(k+j|k)=\left[\sum_{q\in\{1,\dots,p-1\}}\mathbf{z}_{cq}^*(k+j|k)\right]+\left[\sum_{q\in\{p+1,\dots,N_p\}}\mathbf{z}_{cq}^*(k+j|k-1)\right]\quad\forall j\in\{0,\dots,N-2\} \tag{19a}$$

$$\tilde{\mathbf{z}}_{cp}(k+N-1|k)=\left[\sum_{q\in\{1,\dots,p-1\}}\mathbf{z}_{cq}^*(k+N-1|k)\right]+\left[\sum_{q\in\{p+1,\dots,N_p\}}\left(\mathbf{E}_{cq}+\mathbf{F}_{cq}\mathbf{K}_q(N-1)\right)\times\mathbf{x}_q^*(k+N-1|k-1)\right], \tag{19b}$$

where $\mathbf{z}_{cq}^*(k+j|k)$ denotes the outputs associated with the optimal solution to sub-problem q at time k. The predicted outputs for $q>p$ are constructed from the remainder of the plan from the previous time-step and a single step of the control law $\mathbf{u}_q=\mathbf{K}_q(N-1)\mathbf{x}_q$, to keep the subsystem state in its terminal set.
- the modified coupling constraint sets $\mathcal{Z}_{cp}(j)$ are given by

$$\forall c\in\{1,\dots,N_c\}:\mathcal{Z}_{cN_p}(0)=\mathcal{Z}_c \tag{20a}$$

$$\mathcal{Z}_{c(p-1)}(j)=\mathcal{Z}_{cp}(j)\sim(\mathbf{E}_{cp}+\mathbf{F}_{cp}\mathbf{K}_p(j))\mathbf{L}_p(j)\mathcal{W}_p \tag{20b}$$

$$\mathcal{Z}_{cN_p}(j+1)=\mathcal{Z}_{c1}(j)\sim(\mathbf{E}_{c1}+\mathbf{F}_{c1}\mathbf{K}_1(j))\mathbf{L}_1(j)\mathcal{W}_1\quad\forall j\in\{0,\dots,N-2\}. \tag{20c}$$

Besides the change to consider only one subsystem, the key difference between the DMPC optimization above and the centralized optimization in § 3 is the tightening of the constraints in (20). Additional margin is included in the coupling constraints of each sub-problem to account for the uncertainty in the predicted plans for other subsystems. It is this feature that ensures the feasibility of the distributed, sequential optimization process. The sub-problems P_p are employed in the following algorithm.

Algorithm 2: *Distributed MPC.*

- (i) *Set* $k=0$. *Find a solution to the initial centralized problem* $\bar{P}(\bar{\mathbf{x}}(0))$. *If a solution cannot be found, stop.*
- (ii) *Apply control* $\mathbf{u}_p(k)=\mathbf{u}_p^*(k|k)$ *to each subsystem p*
- (iii) *Increment k.*
- (iv) *For each subsystem p in order* $1,\dots,N_p$:
    - (a) *Construct the outputs* $\tilde{\mathbf{Z}}_p(k)$ *from other subsystems, defined by* (19), *via communication.*
    - (b) *Solve sub-problem* $P_p(\mathbf{x}_p(k),\tilde{\mathbf{Z}}_p(k))$.
- (v) *Go to Step* (ii).

Note that Step (i) does not require finding the optimal solution of the centralized problem, although that is one way of finding the initial plan. Also, the algorithm presented above implies communication between all pairs of subsystems at every step, to provide the information (19) required in Step (iv)a. This form will be retained throughout the next section for simplicity when proving the properties of the algorithm. Then in § 6, it is shown that the identification of structure in the constraints leads to equivalent forms with much weaker communication requirements.

In Algorithm 2, observe that the state $\mathbf{x}_p(k)$ of subsystem p is not needed until the solution of sub-problem $P_p(\mathbf{x}_p(k),\tilde{\mathbf{Z}}_p(k))$, and that the control $\mathbf{u}_p(k)$ is available as soon as that sub-problem is solved. Therefore, by judicious choice of sampling schemes (Richards and How 2005), the computational delay in each local feedback loop is only the time to solve the local sub-problem in Step (iv)b, not the whole sequence of sub-problems in Step (iv).

5. Properties of DMPC

This section proves the main results of this paper: that Algorithm 2 guarantees feasibility of all optimizations involved, satisfaction of constraints (4) and (6), and convergence to the targets (8).

5.1 *Feasibility and constraint satisfaction*

The development for feasibility and constraint satisfaction involves several intermediate results, all proving that the feasibility of each sub-problem follows from feasibility of its predecessor in Algorithm 2, as shown in figure 1(a). This sequence is proven using three propositions. Proposition 1 shows that existence of a feasible solution to the centralized problem at some time step implies feasibility of the first sub-problem at the subsequent time step, Proposition 2 shows that feasibility of any sub-problem 1 to (N_p-1) at any step implies feasibility of the subsequent sub-problem; and Proposition 3 shows that feasibility of the final sub-problem N_p is equivalent to the existence of a feasible solution to the centralized problem. The overall result follows by recursion: the existence of an initial feasible centralized solution, ensured by Step (i) of Algorithm 2, implies feasibility of all subsequent optimizations.

In every case, feasibility is shown by constructing a particular candidate solution, using the candidate policies $\mathbf{K}_p(j)$ to compensate for disturbances. The section begins with a lemma showing that the constraints of one of the sub-problems are equivalent to those of the centralized optimization in § 3, which will be useful in subsequent results. There follow the three propositions forming the overall recursion. The section ends with the theorem combining the intermediate results to prove the main result.

Lemma 1: $\mathcal{Z}_{cN_p}(j)=\bar{\mathcal{Z}}_c(j)\ \forall j\in\{0,\dots,N\}$, *i.e., the constraint sets for the summed coupling outputs in the centralized problem* $\bar{P}$, *computed using* (13), *are identical to the constraint sets for the coupling outputs in sub-problem* P_{N_p}, *computed using* (20).

Proof: Comparison of (13a) and (20a) shows that $\mathcal{Z}_{cN_p}(0)=\bar{\mathcal{Z}}_c(0)$. From Kolmanovsky and Gilbert (1998), Theorem 2.1(v), it is known that

$$(\mathcal{A}\sim\mathcal{B}_1)\sim\mathcal{B}_2=\mathcal{A}\sim(\mathcal{B}_1\oplus\mathcal{B}_2).$$

Applying this result to the recursions in (20b) gives

$$\begin{aligned}\mathcal{Z}_{c1}(j)=\mathcal{Z}_{cN_p}(j)\sim\big[&(\mathbf{E}_{c2}+\mathbf{F}_{c2}\mathbf{K}_2(j))\mathbf{L}_2(j)\mathcal{W}_2\oplus\cdots\\&\oplus(\mathbf{E}_{cN_p}+\mathbf{F}_{cN_p}\mathbf{K}_{N_p}(j))\mathbf{L}_{N_p}(j)\mathcal{W}_{N_p}\big].\end{aligned}$$

Appending the recursion step (20c) then gives

$$\begin{aligned}\mathcal{Z}_{cN_p}(j+1)=\mathcal{Z}_{cN_p}(j)\sim\big[&(\mathbf{E}_{c1}+\mathbf{F}_{c1}\mathbf{K}_1(j))\mathbf{L}_1(j)\mathcal{W}_1\oplus\cdots\\&\oplus(\mathbf{E}_{cN_p}+\mathbf{F}_{cN_p}\mathbf{K}_{N_p}(j))\mathbf{L}_{N_p}(j)\mathcal{W}_{N_p}\big]\end{aligned}$$

which is identical to the recursion (13c) used to construct sets $\bar{\mathcal{Z}}_c(j)$. Hence, since the initial sets $\mathcal{Z}_{cN_p}(0)$ and $\bar{\mathcal{Z}}_c(0)$ are identical and the recursions used to construct subsequent sets $\mathcal{Z}_{cN_p}(j)$ and $\bar{\mathcal{Z}}_c(j)$ for $j > 0$ have been shown to be equivalent, then $\mathcal{Z}_{cN_p}(j) = \bar{\mathcal{Z}}_c(j)$ for all j. □

Notation: In the remainder of this section, the following notation is employed.

- $\bar{\mathbf{U}}^*(k)$ denotes a known solution to the centralized problem $\bar{P}(\bar{\mathbf{x}}(k))$.
- $\mathbf{U}_p^*(k)$ denotes a known solution to the sub-problem $P_p(\mathbf{x}_p(k), \tilde{\mathbf{Z}}_p(k))$.
- $\mathbf{u}_p^*(\cdot|k)$ denotes the individual control elements of the known solution $\bar{\mathbf{U}}^*(k)$.
- $\mathbf{x}_p^*(\cdot|k)$, $\mathbf{y}_p^*(\cdot|k)$, $\mathbf{z}_{cp}^*(\cdot|k)$ and $\mathbf{s}_p^*(\cdot|k)$ denote the individual state, local constraint output, coupling constraint output and tracking output elements, respectively, associated with the known solution $\mathbf{U}_p^*(k)$, together satisfying (18b), (18c) (18d) and (18e).
- $\hat{\mathbf{U}}_p(k)$ denotes a candidate solution to the sub-problem $P_p(\mathbf{x}_p(k), \tilde{\mathbf{Z}}_p(k))$.
- $\hat{\mathbf{u}}_p(\cdot|k)$ denotes the individual control elements of $\hat{\mathbf{U}}_p(k)$.
- $\hat{\mathbf{x}}_p(\cdot|k)$, $\hat{\mathbf{y}}_p(\cdot|k)$, $\hat{\mathbf{z}}_{cp}(\cdot|k)$ and $\hat{\mathbf{s}}_p(\cdot|k)$ denote the individual state, local constraint output, coupling constraint output and tracking output elements, respectively, associated with the candidate solution $\hat{\mathbf{U}}_p(k)$, together satisfying (18b), (18c) (18d) and (18e).

The following three propositions are required to prove feasibility of Algorithm 2. Note that each proposition follows on from its predecessor. This sequence will also be used in a recursion in the subsequent Theorem to prove that feasibility of each sub-problem follows from feasibility of the preceding sub-problem.

Proposition 1: *If*

(i) *at any time step k_0, a set of solutions $\bar{\mathbf{U}}^*(k_0)$ is known, satisfying the constraints of $\bar{P}(\bar{\mathbf{x}}(k_0))$ in* § 3, *and*

(ii) *the controls $\mathbf{u}_p(k_0) = \mathbf{u}_p^*(k_0|k_0)$ are applied to the system at time step k_0,*

then at the subsequent time step k_0+1 the first sub-problem $P_1(\mathbf{x}_1(k_0+1), \tilde{\mathbf{Z}}_1(k_0+1))$ in § 4 *is feasible for all disturbances $\mathbf{w}_1(k_0) \in \mathcal{W}_1$.*

Proof: Consider the following candidate solution $\hat{\mathbf{U}}_1(k_0+1)$ for sub-problem $P_1(\mathbf{x}_1(k_0+1), \tilde{\mathbf{Z}}_1(k_0+1))$

$$\begin{aligned}\hat{\mathbf{u}}_1(k_0 + 1 + j|k_0 + 1) &= \mathbf{u}_1^*(k_0 + 1 + j|k_0) + \mathbf{K}_1(j)\mathbf{L}_1(j)\mathbf{w}_1(k_0) \\ &\quad \forall j \in \{0, \dots, N\text{–}2\} \qquad (21\text{a})\end{aligned}$$

$$\begin{aligned}\hat{\mathbf{u}}_1(k_0 + N|k_0 + 1) &= \mathbf{K}_1(N-1)\hat{\mathbf{x}}_1(k_0 + N|k_0 + 1) \qquad (21\text{b})\end{aligned}$$

This candidate solution is formed by taking from the tail of the known feasible solution $\bar{\mathbf{U}}^*(k_0)$, adding one step using controller $\mathbf{K}_1(N-1)$ at the end, and adding perturbations representing the rejection of the disturbance by the candidate controller $\mathbf{K}_1(\cdot)$ and the associated state transition matrices $\mathbf{L}_1(\cdot)$.

The initial state $\mathbf{x}_1(k_0+1)$ is given by the true dynamics $\mathbf{x}_1(k_0+1) = \mathbf{A}_1\mathbf{x}_1(k_0) + \mathbf{B}_1\mathbf{u}_1(k_0) + \mathbf{w}_1(k_0)$. Also, the dynamics constraints of $\bar{P}(\bar{\mathbf{x}}(k_0))$ ensure $\mathbf{x}_1^*(k_0+1|k_0) = \mathbf{A}_1\mathbf{x}_1(k_0) + \mathbf{B}_1\mathbf{u}_1(k_0)$. Comparing these two expressions shows $\mathbf{x}_1(k_0+1) = \mathbf{x}_1^*(k_0+1|k_0) + \mathbf{w}_1(k_0)$. With this initial condition and the candidate control sequence (21a), the dynamics constraint (9b) gives the following candidate state sequence, also expressed as a perturbation from the plan at time k_0 and using the state transition matrices (16)

$$\begin{aligned}\hat{\mathbf{x}}_1(k_0 + 1 + j|k_0 + 1) &= \mathbf{x}^*(k_0 + j + 1|k_0) + \mathbf{L}_1(j)\mathbf{w}_1(k_0) \\ &\quad \forall j \in \{0, \dots, N-1\} \qquad (22)\end{aligned}$$

and corresponding local constraint output sequences

$$\begin{aligned}\hat{\mathbf{y}}_1(k_0 + 1 + j|k_0 + 1) &= \mathbf{y}_1^*(k_0 + j + 1|k_0) \\ &\quad + (\mathbf{C}_1 + \mathbf{D}_1\mathbf{K}_1(j))\mathbf{L}_1(j)\mathbf{w}_1(k_0) \\ &\quad \forall j \in \{0, \dots, N-2\} \qquad (23\text{a}) \\ \hat{\mathbf{y}}_1(k_0 + N|k_0 + 1) &= (\mathbf{C}_1 + \mathbf{D}_1\mathbf{K}_1(N-1)) \\ &\quad \times \mathbf{x}_1^*(k_0 + N|k_0) \\ &\quad + (\mathbf{C}_1 + \mathbf{D}_1\mathbf{K}_1(N-1)) \\ &\quad \times \mathbf{L}_1(N-1)\mathbf{w}_1(k_0). \qquad (23\text{b})\end{aligned}$$

Feasibility of $\bar{\mathbf{U}}^*(k_0)$ implies $\mathbf{y}_1^*(k_0+j+1|k_0) \in \mathcal{Y}_1(j+1)\ \forall j \in \{0, \dots, N-1\}$, from (9g), and $(\mathbf{C}_1 + \mathbf{D}_1\mathbf{K}_1(N-1))\mathbf{x}_1^*(k_0+N|k_0) \in \mathcal{Y}_1(N)$, from (9f) and (17b). Therefore, since $\mathcal{Y}_1(j+1)$ is related to $\mathcal{Y}_1(j)$ by the Pontryagin difference in (11b) with the known property (15), the perturbation expressions (23) imply $\hat{\mathbf{y}}_1(k_0+1+j|k_0+1) \in \mathcal{Y}_1(j)\ \forall j \in \{0, \dots, N-1\}$, satisfying all the local constraints.

Next, consider the coupling constraints (18h). Similar to (23), the coupling constraint outputs associated with the candidate solution (21a) are

$$\begin{aligned}&\forall c \in \{1, \dots, N_c\}: \\ \hat{\mathbf{z}}_{c1}(k_0 + 1 + j|k_0 + 1) &= \mathbf{z}_{c1}^*(k_0 + j + 1|k_0) \\ &\quad + (\mathbf{E}_{c1} + \mathbf{F}_{c1}\mathbf{K}_1(j))\mathbf{L}_1(j)\mathbf{w}_1(k_0) \\ &\quad \forall j \in \{0, \dots, N-2\} \qquad (24\text{a}) \\ \hat{\mathbf{z}}_{c1}(k_0 + N|k_0 + 1) &= (\mathbf{E}_{c1} + \mathbf{F}_{c1}\mathbf{K}_1(N-1)) \\ &\quad \times \mathbf{x}^*(k_0 + N|k_0) \\ &\quad + (\mathbf{E}_{c1} + \mathbf{F}_{c1}\mathbf{K}_1(N-1)) \\ &\quad \times \mathbf{L}_1(N-1)\mathbf{w}_1(k_0) \qquad (24\text{b})\end{aligned}$$

Combining these with the communicated output sequences $\tilde{\mathbf{Z}}_1(k_0+1)$, constructed using (19), gives expressions for the summation on the left hand side of (18h)

$$\hat{\mathbf{z}}_{c1}(k_0+1+j|k_0+1)+\tilde{\mathbf{z}}_{c1}(k_0+1+j|k_0+1)$$
$$=\left[\sum_{q\in\{1\ldots N_p\}}\mathbf{z}^*_{cq}(k_0+1+j|k_0)\right]$$
$$+(\mathbf{E}_{c1}+\mathbf{F}_{c1}\mathbf{K}_1(j))\mathbf{L}_1(j)\mathbf{w}_1(k_0)$$
$$\forall j\in\{0,\ldots,N-2\} \tag{25a}$$
$$\hat{\mathbf{z}}_{c1}(k_0+N|k_0+1)+\tilde{\mathbf{z}}_{c1}(k_0+N|k_0+1)$$
$$=\left[\sum_{q\in\{1\ldots N_p\}}(\mathbf{E}_{c1}+\mathbf{F}_{c1}\mathbf{K}_1(N-1))\mathbf{x}^*(k_0+N|k_0)\right]$$
$$+(\mathbf{E}_{c1}+\mathbf{F}_{c1}\mathbf{K}_1(N-1))\mathbf{L}_1(N-1)\mathbf{w}_1(k_0). \tag{25b}$$

In order to show that these quantities satisfy the constraints (9h), first use the fact that since $\bar{\mathbf{U}}^*(k_0)$ satisfies the constraints of $\bar{P}(\bar{\mathbf{x}}(k_0))$ then from (9h), (9f) and (17c)

$$\left[\sum_{q\in\{1\ldots N_p\}}\mathbf{z}^*_{cq}(k_0+1+j|k_0)\right]\in\bar{\mathcal{Z}}_c(j+1),$$
$$\forall j\in\{0,\ldots,N-2\}$$
$$\left[\sum_{q\in\{1\ldots N_p\}}(\mathbf{E}_{c1}+\mathbf{F}_{c1}\mathbf{K}_1(N-1))\mathbf{x}^*(k_0+N|k_0)\right]\in\bar{\mathcal{Z}}_c(N).$$

Then, using $\bar{\mathcal{Z}}_c(j)=\mathcal{Z}_{cN_p}(j)\ \forall j$ from Lemma 1, the Pontryagin differences (20c) and the perturbation expressions (25) show $\hat{\mathbf{z}}_{c1}(k_0+j+1|k_0+1)+\tilde{\mathbf{z}}_{c1}(k_0+j+1|k_0+1)\in\mathcal{Z}_{c1}(j)\ \forall j\in\{0,\ldots,N-1\}$. Hence the coupling constraints are satisfied.

Finally, it remains to show that $\hat{\mathbf{U}}_p(k_0+1)$ satisfies the terminal constraints. Combining the final control step (21b) and the state sequence (22) in the dynamics constraint (18b) gives the predicted terminal state

$$\hat{\mathbf{x}}_1(k_0+N+1|k_0+1)=(\mathbf{A}_1+\mathbf{B}_1\mathbf{K}_1(N-1))$$
$$\times\,\mathbf{x}^*_1(k_0+N|k_0)$$
$$+(\mathbf{A}_1+\mathbf{B}_1\mathbf{K}_1(N-1))$$
$$\times\,\mathbf{L}_1(N-1)\mathbf{w}_1(k_0).$$

Feasibility of $\bar{\mathbf{U}}^*(k_0)$ implies $\mathbf{x}^*_1(k_0+N|k_0)\in\mathcal{T}_1$ and hence the invariance requirement (17a) ensures $\hat{\mathbf{x}}_1(k_0+N+1|k_0+1)\in\mathcal{T}_1$, satisfying the terminal constraint.

The above has shown that the candidate sequence $\hat{\mathbf{U}}_1(k_0+1)$ is a feasible solution to $P_1(\mathbf{x}_1(k_0+1),\tilde{\mathbf{Z}}_1(k_0+1))$, hence this sub-problem must be feasible. □

Proposition 2: *If*

(i) *at any time step k_0, a set of solutions $\bar{\mathbf{U}}^*(k_0)$ is known, satisfying the constraints of $\bar{P}(\bar{\mathbf{x}}(k_0))$,*
(ii) *the controls $\mathbf{u}_p(k_0)=\mathbf{u}^*_p(k_0|k_0)$ are applied to the system at time step k_0,*
(iii) *for some subsystem $p_0\in\{1,\ldots,(N_p-1)\}$ a solution $\mathbf{U}^*_{p_0}(k_0+1)$ is known, satisfying the constraints of sub-problem $P_{p_0}(\mathbf{x}_{p_0}(k_0+1),\tilde{\mathbf{Z}}_{p_0}(k_0+1))$,*

then the sub-problem $P_{p_0+1}(\mathbf{x}_{p_0+1}(k_0+1),\tilde{\mathbf{Z}}_{p_0+1}(k_0+1))$ is feasible for all disturbances $\mathbf{w}_{p_0+1}(k_0)\in\mathcal{W}_{p_0+1}$.

Proof: Consider the following candidate solution $\hat{\mathbf{U}}_{p_0+1}(k_0+1)$ for $P_{p_0+1}(\mathbf{x}_{p_0+1}(k_0+1),\tilde{\mathbf{Z}}_{p_0+1}(k_0+1))$

$$\hat{\mathbf{u}}_{p_0+1}(k_0+1+j|k_0+1)=\mathbf{u}^*_{p_0+1}(k_0+1+j|k_0)$$
$$+\mathbf{K}_{p_0+1}(j)\mathbf{L}_{p_0+1}(j)\mathbf{w}_{p_0+1}(k_0)$$
$$\forall j\in\{0,\ldots,N-2\} \tag{26a}$$
$$\hat{\mathbf{u}}_{p_0+1}(k_0+N|k_0+1)=\mathbf{K}_{p_0+1}(N-1)$$
$$\times\,\hat{\mathbf{x}}_{p_0+1}(k_0+N|k_0+1). \tag{26b}$$

This sequence is constructed in the same way as the candidate in (21). Satisfaction of the local constraints (18g) and the terminal constraints (18f) follow from identical arguments to those used in Proposition 1. It remains to show that the coupling constraints (18h) are satisfied by this candidate. Substituting the candidate sequence $\hat{\mathbf{U}}_{p_0+1}(k_0+1)$ into the construction of the communicated data (19) gives

$$\hat{\mathbf{z}}_{c(p_0+1)}(k_0+j+1|k_0+1)+\tilde{\mathbf{z}}_{c(p_0+1)}(k_0+j+1|k_0+1)$$
$$=\mathbf{z}^*_{cp_0}(k_0+j+1|k_0+1)+\tilde{\mathbf{z}}_{cp_0}(k_0+j+1|k_0+1)$$
$$+\left(\mathbf{E}_{c(p_0+1)}+\mathbf{F}_{c(p_0+1)}\mathbf{K}_{p_0+1}(j)\right)\mathbf{L}_{p_0+1}(j)\mathbf{w}_{p_0+1}(k_0),$$
$$\forall j\in\{0,\ldots,N-1\} \tag{27}$$

and we know from feasibility of $\mathbf{U}^*_{p_0}(k_0+1)$ that

$$\mathbf{z}^*_{cp_0}(k_0+j+1|k_0+1)+\tilde{\mathbf{z}}_{cp_0}(k_0+j+1|k_0+1)$$
$$\in\mathcal{Z}_{cp_0}(j)\quad\forall j\in\{0,\ldots,N-1\}.$$

Therefore, by applying the property of the Pontryagin difference (15) to the constraint tightening recursion (20b), we find that

$$\hat{\mathbf{z}}_{c(p_0+1)}(k_0+j+1|k_0+1)+\tilde{\mathbf{z}}_{c(p_0+1)}(k_0+j+1|k_0+1)$$
$$\in\mathcal{Z}_{c(p_0+1)}$$

hence the coupling constraints (18h) are satisfied, the candidate $\hat{\mathbf{U}}_{p_0+1}(k_0+1)$ is feasible and so the sub-problem $P_{p_0+1}(\mathbf{x}_{p_0+1}(k_0+1), \tilde{\mathbf{Z}}_{p_0+1}(k_0+1))$ is feasible. □

Proposition 3: *If at any time step k_0+1 there exists a solution $\mathbf{U}_p^*(k_0+1)$ for every $p \in \{1,\dots,N_p\}$, each satisfying the constraints of corresponding sub-problem $P_p(\mathbf{x}_p(k_0+1), \tilde{\mathbf{Z}}_p(k_0+1))$ then the combination of those solutions $\bar{\mathbf{U}}(k_0+1) = \{\mathbf{U}_1^*(k_0+1),\dots,\mathbf{U}_{N_p}^*(k_0+1)\}$ also satisfies the constraints of the centralized problem $\bar{P}(\bar{\mathbf{x}}(k_0+1))$.*

Proof: Satisfaction of the local constraints (9g) and terminal constraints (9f) of the centralized problem follow from satisfaction of the corresponding constraints (18g) and (18f) of each sub-problem $p \in \{1,\dots,N_p\}$. It remains to consider the coupling constraints (9h). Substituting into (18h) with $p = N_p$ for the communicated outputs from (19) gives

$$\sum_{p=1}^{N_p} \mathbf{z}_{cp}^*(k_0+j+1|k_0+1) \in \mathcal{Z}_{cN_p},$$

which, since $\bar{\mathcal{Z}}_c(j) = \mathcal{Z}_{cN_p}(j)\ \forall j$ from Lemma 1, implies that the coupling constraints (9h) of the centralized problem are satisfied if the coupling constraints (18h) of the final sub-problem $P_{N_p}(\mathbf{x}_{N_p}(k_0+1), \tilde{\mathbf{Z}}_{N_p}(k_0+1))$ are satisfied. Therefore $\bar{\mathbf{U}}(k_0+1)$ satisfies all the constraints of the centralized problem $\bar{P}(\bar{\mathbf{x}}(k_0+1))$.

Theorem 1: Robust feasibility and constraint satisfaction of DMPC. *If the systems* (1) *are subjected to disturbances obeying* (2) *and controlled using Algorithm* 2, *and if a feasible solution to the initial centralized problem* $\bar{P}(\bar{\mathbf{x}}(0))$ *can be found, then all subsequent sub-problems* $P_p(\mathbf{x}_p(k), \tilde{\mathbf{Z}}_p(k))$ *are feasible and the outputs* (3) *and* (5) *satisfy constraints* (4) *and* (6), *respectively.*

Proof: Assume that at some time step k_0, a solution is known to the centralized problem $\bar{P}(\bar{\mathbf{x}}(k_0))$. Then by Proposition 1, sub-problem $P_1(\mathbf{x}_1(k_0+1), \tilde{\mathbf{Z}}_1(k_0+1))$ is feasible. Using Proposition 2 recursively, every subsequent sub-problem $P_p(\mathbf{x}_p(k_0+1), \tilde{\mathbf{Z}}_p(k_0+1))$ is feasible for $p \in \{2,\dots,N_p\}$. By Proposition 3, feasibility of the final sub-problem $P_{N_p}(\mathbf{x}_{N_p}(k_0+1), \tilde{\mathbf{Z}}_{N_p}(k_0+1))$ implies that a solution is known for the centralized problem $\bar{P}(\bar{\mathbf{x}}(k_0+1))$. This completes an outer recursion, showing that knowledge of a feasible solution to $\bar{P}(\bar{\mathbf{x}}(k_0))$ implies that a feasible solution can be found for $\bar{P}(\bar{\mathbf{x}}(k_0+1))$. Therefore, knowledge of an initial feasible solution $\bar{\mathbf{U}}(0)$ implies feasibility of every subsequent sub-problem in the DMPC algorithm.

Constraint satisfaction follows from Proposition 3, which showed that the combined solutions to all the sub-problems satisfied the centralized constraints. Substituting $j = 0$ into (9) and combining with the control law $\mathbf{u}_p(k) = \mathbf{u}_p^*(k|k)$ yields constraints that are equivalent to those in the problem statements (4) and (6). Therefore feasibility implies constraint satisfaction. □

5.2 *Convergence to target*

The proof of convergence under Algorithm 1 builds on the results concerning feasibility in §5.1. Propositions 1, 2 and 3 showed that a particular candidate solution was feasible for each sub-problem. To prove convergence, this solution is used to provide an upper bound on the optimal cost $J_p^*(\mathbf{x}_p(k), \tilde{\mathbf{Z}}_p(k))$ for each sub-problem. This then leads to a Lyapunov-like result showing proving the desired convergence result (8). The section begins with a lemma regarding the distance metric which will be useful in the proof of robust convergence.

Lemma 2: *Property of distance metric. Let $\mathcal{B}$ and $\mathcal{C}$ be compact sets. For any* $\mathbf{a}$, $d(\mathbf{a}+\mathbf{c}, \mathcal{B}) \le d(\mathbf{a}, \mathcal{B} \sim \mathcal{C})\ \forall \mathbf{c} \in \mathcal{C}$.

Proof: Let $d_1 = d(\mathbf{a}, \mathcal{B} \sim \mathcal{C})$. From the definition (10) of the metric, there exists a $\mathbf{b}^* \in \mathcal{B} \sim \mathcal{C}$ such that $\|\mathbf{a} - \mathbf{b}^*\| = d_1$. Then by the definition (14) of the Pontryagin difference, the vector $\mathbf{b}^* + \mathbf{c} \in \mathcal{B}$ for all $\mathbf{c} \in \mathcal{C}$. Therefore $d(\mathbf{a}+\mathbf{c}, \mathcal{B}) = \min_{b \in \mathcal{B}} \|\mathbf{a}+\mathbf{c}-\mathbf{b}\| \le \|\mathbf{a}+\mathbf{c}-(\mathbf{b}^*+\mathbf{c})\| = \|\mathbf{a}-\mathbf{b}^*\| = d_1$. □

Theorem 2: *Robust convergence of DMPC. If the systems* (1) *are subjected to disturbances obeying* (2) *and controlled using Algorithm* 2, *and if a feasible solution to the initial centralized problem* $\bar{P}(\bar{\mathbf{x}}(0))$ *can be found, then the tracking outputs* (7) *converge as required by* (8).

Proof: From Theorem 1 and Propositions 1 and 2, we know that if $U_p^*(k_0)$ is a feasible (in this case also optimal) solution to $P_p(\mathbf{x}(k_0), \tilde{\mathbf{Z}}_p(k_0))$ for any $k_0 \ge 1$ and the control $\mathbf{u}_p^*(k_0) = \mathbf{u}_p^*(k_0|k_0)$ is applied, then a candidate solution $\hat{\mathbf{U}}_p(k_0+1)$ is feasible solution to sub-problem $P_p(\mathbf{x}_p(k_0+1), \tilde{\mathbf{Z}}_p(k_0+1))$ for $p \in \{1,\dots,N_p\}$ where the control elements of the candidate are given by

$$\begin{aligned}\hat{\mathbf{u}}_p(k_0+1+j|k_0+1) &= \mathbf{u}_p^*(k_0+1+j|k_0)\\ &\quad + \mathbf{K}_p(j)\mathbf{L}_p(j)\mathbf{w}_p(k_0)\\ &\quad \forall j \in \{0,\dots,N-2\}\\ \hat{\mathbf{u}}_p(k_0+N|k_0+1) &= \mathbf{K}_p(N-1)\hat{\mathbf{x}}_p(k_0+N|k_0+1).\end{aligned} \tag{28}$$

Then from (18e), the predicted tracking outputs associated with that candidate solution are

$$\begin{aligned}\hat{\mathbf{s}}_p(k_0+j+1|k_0+1) &= \mathbf{s}_p^*(k_0+j+1|k_0)\\ &\quad + (\mathbf{G}_p+\mathbf{H}_p\mathbf{K}_p(j))\mathbf{L}_p(j)\mathbf{w}_p(k_0)\\ &\qquad \forall j\in\{0,\dots,N-2\} \qquad (29a)\\ \hat{\mathbf{s}}_p(k_0+N|k_0+1) &= \big(\mathbf{G}_p+\mathbf{H}_p\mathbf{K}_p(N-1)\big)\\ &\quad \times \mathbf{x}_p^*(k_0+N|k_0+1)\\ &\quad + \big(\mathbf{G}_p+\mathbf{H}_p\mathbf{K}_p(N-1)\big)\\ &\quad \times \mathbf{L}_p(N-1)\mathbf{w}_p(k_0). \qquad (29b)\end{aligned}$$

Applying Lemma 2 to (29a) gives

$$\begin{aligned}&d(\hat{\mathbf{s}}_p(k_0+1+j|k_0+1), \mathcal{S}_p(j))\\ &\quad \le d\big(\mathbf{s}_p^*(k_0+j+1|k_0),\\ &\qquad \mathcal{S}_p(j) \sim (\mathbf{G}_p+\mathbf{H}_p\mathbf{K}_p(j))\mathbf{L}_p(j)\mathcal{W}_p\big)\\ &\qquad\quad \forall j\in\{0,\dots,N-2\},\ \forall \mathbf{w}_p(k_0)\in\mathcal{W}_p\end{aligned}$$

and by comparison with the target set tightening (12b) this is equivalent to

$$\begin{aligned}&d\big(\hat{\mathbf{s}}_p(k_0+1+j|k_0+1), \mathcal{S}_p(j)\big)\\ &\quad \le d\big(\mathbf{s}_p^*(k_0+j+1|k_0), \mathcal{S}_p(j+1)\big) \quad \forall j\in\{0,\dots,N-2\},\\ &\quad \forall \mathbf{w}_p(k_0)\in\mathcal{W}_p.\end{aligned}$$

Furthermore, feasibility of $\mathbf{U}_p^*(k_0)$ implies $(\mathbf{G}_p+\mathbf{H}_p\mathbf{K}_p(N-1))\mathbf{x}_p^*(k_0+N|k_0)\in\mathcal{S}_p(N)$, from (9f) and (17d). Therefore, the Pontryagin difference in (12b) and the property (15) imply $\hat{\mathbf{s}}_p(k_0+N|k_0+1)\in\mathcal{S}_p(N-1)$, such that the final step incurs no cost because the predicted tracking output is inside the target set

$$d\big(\hat{\mathbf{s}}_p(k_0+N|k_0+1), \mathcal{S}_p(N-1)\big) = 0.$$

Combining these findings gives a bound on the cost $J_p(\mathbf{x}_p(k_0+1), \tilde{\mathbf{Z}}_p(k_0+1), \hat{\mathbf{U}}_p(k_0+1))$

$$\begin{aligned}&J_p\Big(\mathbf{x}_p(k_0+1), \tilde{\mathbf{Z}}_p(k_0+1), \hat{\mathbf{U}}_p(k_0+1)\Big)\\ &\quad = \sum_{j=0}^{N-1} d\big(\hat{\mathbf{s}}_p(k_0+j+1|k_0+1), \mathcal{S}_p(j)\big)\\ &\quad \le \sum_{j=0}^{N-2} d\Big(\mathbf{s}_p^*(k_0+j+1|k_0), \mathcal{S}_p(j+1)\Big)\end{aligned}$$

and this bound can be expressed in terms of the optimal cost at k_0

$$\begin{aligned}&J_p\Big(\mathbf{x}_p(k_0+1), \tilde{\mathbf{Z}}_p(k_0+1), \hat{\mathbf{U}}_p(k_0+1)\Big)\\ &\quad \le J_p^*\Big(\mathbf{x}_p(k_0), \tilde{\mathbf{Z}}_p(k_0)\Big) - d\Big(\mathbf{s}_p^*(k_0|k_0),\ \mathcal{S}_p(0)\Big).\end{aligned}$$

Since $\hat{\mathbf{U}}_p(k_0+1)$ is a feasible solution, this provides a bound on the optimal cost

$$\begin{aligned}J_p^*\Big(\mathbf{x}(k_0+1), \tilde{\mathbf{Z}}_p(k_0+1)\Big) &\le J_p^*\Big(\mathbf{x}_p(k_0), \tilde{\mathbf{Z}}_p(k_0)\Big)\\ &\quad - d(\mathbf{s}_p^*(k_0|k_0), \mathcal{S}_p(0)\Big)\end{aligned}$$

Since $d(\cdot)\ge 0$ by construction in (10), then $J_p^*(\mathbf{x}_p(k), \tilde{\mathbf{Z}}_p(k))$ must be a decreasing function. Furthermore, $J_p^*(\mathbf{x}_p(k), \tilde{\mathbf{Z}}_p(k))$ cannot become negative, by definition, so it must converge to a steady value. This implies $d(\mathbf{s}_p^*(k|k), \mathcal{S}_p(0)) = d(\mathbf{s}_p(k), \mathcal{S}_p) \to 0$ as $k\to\infty$ which, by the definition of the metric $d(\cdot)$ and the assumption that each $\mathcal{S}_p$ is compact, implies that $\mathbf{s}_p(k)\to\mathcal{S}_p$ as $k\to\infty$.

Remark 1: Note that the results in § 5.1 did not make use of the cost function or of optimality of the solutions. Therefore, for problems that do not require convergence and are only concerned with constraint satisfaction, any form of cost function can be employed and the solutions found for each sub-problem need only to be feasible, not necessarily optimal. For example, spacecraft formation control is sometimes expressed as the need for each spacecraft remain in an "error box" relative to the rest of the formation (Inalhan *et al.* 2002). It would be possible to employ DMPC minimizing fuel consumption, the typical objective for spacecraft, and still guarantee constraint satisfaction.

6. Communication requirements

The development so far has assumed that problem $P_p(\mathbf{x}_p(k), \tilde{\mathbf{Z}}_p(k))$ includes coupling data $\mathbf{z}_{cq}^*(k+j|k)$ for all constraints $c\in\{1,\dots,N_c\}$ and for all other subsystems $q\ne p$, according to (19). This implies communication with all other subsystems at every time step. However, this section shows that by identifying structure in the coupling, significant relaxation of that requirement can be achieved.

Define $\mathcal{P}_c$ as the set of all subsystems affected by constraint c

$$\mathcal{P}_c = \big\{p\in\{1,\dots,N_p\}\ |\ [\mathbf{E}_{cp}\ \mathbf{F}_{cp}]\ne\mathbf{0}\big\}. \qquad (30)$$

This allows us to identify and eliminate identically zero terms in the summation of the coupling data (19), leading to the following equivalent form.

$$\tilde{\mathbf{z}}_{cp}(k+j|k) = \left[\sum_{q\in\mathcal{P}_c\cap\{1,\dots,p-1\}} \mathbf{z}^*_{cq}(k+j|k)\right] + \left[\sum_{q\in\mathcal{P}_c\cap\{p+1,\dots,N_p\}} \mathbf{z}^*_{cq}(k+j|k-1)\right] \quad \forall j \in \{0,\dots,N-1\} \tag{31a}$$

$$\tilde{\mathbf{z}}_{cp}(k+N-1|k) = \left[\sum_{q\in\mathcal{P}_c\cap\{1,\dots,p-1\}} \mathbf{z}^*_{cq}(k+N-1|k)\right] + \left[\sum_{q\in\mathcal{P}_c\cap\{p+1,\dots,N_p\}} \left(\mathbf{E}_{cq} + \mathbf{F}_{cq}\mathbf{K}_q(N-1)\right) \times \mathbf{x}^*_q(k+N-1|k-1)\right]. \tag{31b}$$

Also define $\mathcal{C}_p$ as the set of constraints involving subsystem p

$$\mathcal{C}_p = \left\{c \in \{1,\dots,N_c\} \mid [\mathbf{E}_{cp}\ \mathbf{F}_{cp}] \neq \mathbf{0}\right\}. \tag{32}$$

Then, according to (18d), constraints not in $\mathcal{C}_p$ have no effect on sub-problem $P_p(\mathbf{x}_p(k), \tilde{\mathbf{Z}}_p(k))$, and hence can be omitted, replacing (18d) and (18h) with

$$\mathbf{z}_{cp}(k+j|k) = \mathbf{E}_{cp}\mathbf{x}_p(k+j|k) + \mathbf{F}_{cp}\mathbf{u}_p(k+j|k) \quad \forall c \in \mathcal{C}_p \tag{33a}$$

$$\mathbf{z}_{cp}(k+j|k) + \tilde{\mathbf{z}}_{cp}(k+j|k) \in \mathcal{Z}_{cp}(j) \quad \forall c \in \mathcal{C}_p \tag{33b}$$

The modified forms (31) and (33) are equivalent to the originals (19) and (18d, 18h), respectively, so the results on feasibility and convergence still hold.

Finally, inspecting (31) and (33) shows that problem p only involves coupling data from other subsystems $q \in \mathcal{P}_c$ for all $c \in \mathcal{C}_p$. Define the set of all other subsystems coupled to p as

$$\mathcal{Q}_p = \left(\bigcup_{c\in\mathcal{C}_p} \mathcal{P}_c\right) \setminus \{p\}. \tag{34}$$

Therefore, sub-problem $P_p(\mathbf{x}_p(k), \tilde{\mathbf{Z}}_p(k))$ only requires information to be communicated from subsystems in $\mathcal{Q}_p$.

Structure in the constraints also enables parallel solution of sub-problems in some cases. Suppose $p_0 + 1 \notin \mathcal{Q}_{p_0}$ for some subsystem $p_0 \in \{1,\dots,N_p - 1\}$. Then sub-problem P_{p_0+1} does not need any information from sub-problem P_{p_0}, and all the information needed to solve sub-problem P_{p_0+1} is available at the time sub-problem P_{p_0} is started. Therefore, sub-problems P_{p_0} and P_{p_0+1} can be solved in parallel. Consider a typical example in which subsystem 1 is the leader and subsystems 2 and 3 are followers, constrained relative to the leader but not to each other. Then sub-problems P_2 and P_3 can be solved in parallel, giving the planning sequence $\{1, (2\,\&\,3), 1, (2\,\&\,3), \dots\}$. This principle can be extended to enable any number of decoupled subsystems to plan in parallel (Kuwata *et al.* 2006).

7. Examples

This section presents simulation results demonstrating DMPC and comparing its performance, in terms of both the objective value and computation time, with centralized MPC. Further results can be found in Richards and How (2004), illustrating the application of DMPC to multi-aircraft collision avoidance, and in Richards and How (2005), implementing DMPC on robotic vehicles.

The examples in this section involve five identical point masses moving in 1-D and required to remain within a specified distance of each other as they moved from an initial offset to the origin. The dynamics of each subsystem were

$$\forall p \in \{1,\dots,5\}\colon\ \mathbf{A}_p = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},\quad \mathbf{B}_p = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}.$$

A random disturbance of up to 0.1 in magnitude acted upon each subsystem

$$\forall p \in \{1,\dots,5\}\colon\ \mathcal{W}_p = \{\mathbf{w} \in \Re^2 \mid \|\mathbf{w}\|_\infty \le 0.1\}.$$

Local constraints restricted the position $|x_{p,1}| \le 10$, velocity $|x_{p,2}| \le 1$ and control $|u_p| \le 1$. The target was to drive all the masses close to the origin $|x_{p,1}| \le 0.5$ and the initial condition had all masses stationary and collocated with $\mathbf{x}_p = [5\ \ 0]^T\ \forall p \in \{1,\dots,5\}$. The corresponding constraint parameters were $\forall p \in \{1,\dots,5\}$:

$$\mathbf{C}_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix},\quad \mathbf{D}_p = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$\mathcal{Y}_p = \{\mathbf{y} \in \Re^3 \mid |y_1| \le 10,\ |y_2| \le 1,\ |y_3| \le 1\}$$

$$\mathbf{G}_p = \begin{bmatrix} 1 & 0 \end{bmatrix},\ \mathbf{H}_p = \begin{bmatrix} 0 \end{bmatrix},$$

$$\mathcal{S}_p = \{s \in \Re \mid |s| \le 0.5\}.$$

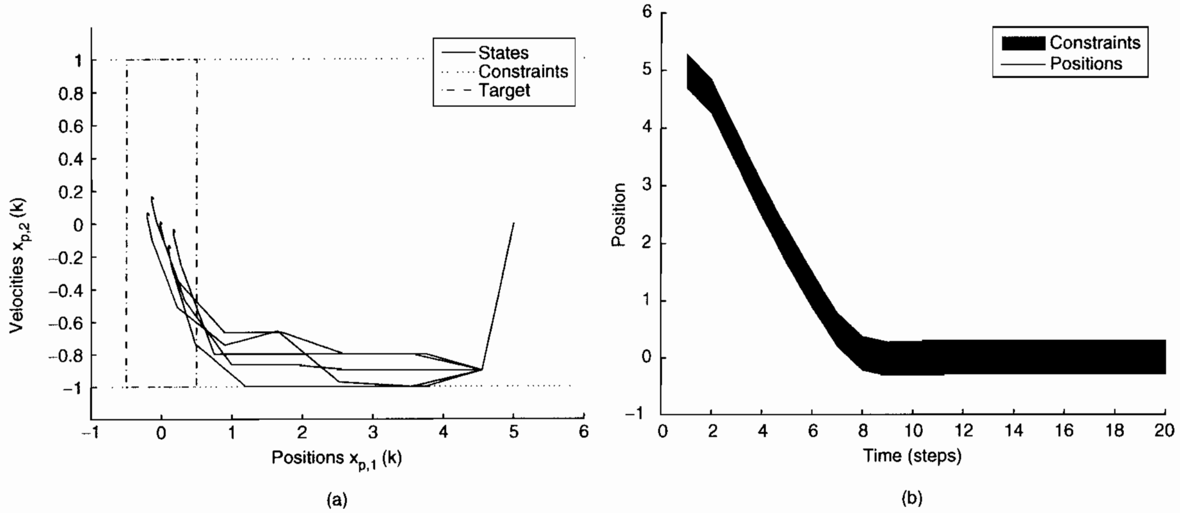


Figure 2. Simulation results from DMPC for example with five point masses. Constraints are satisfied throughout and states converge to targets. (a) State space trajectories; (b) Position time histories.

The distance metrics $d(\cdot,\cdot)$ were implemented using one norms. There were

$$N_c = \binom{5}{2} = 10$$

coupling constraints, requiring that the positions of every pair of masses remain within 0.6. Hence the coupling constraint sets were $\mathcal{Z}_c = \{z \in \Re \mid |z| \leq 0.6\}$ for all c and, for the constraint c_{pq} associated with the pair (p, q), the coupling output matrices were

$$\mathbf{E}_{c_{pq}p} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \ \mathbf{E}_{c_{pq}q} = -\mathbf{E}_{c_{pq}p},$$
$$\mathbf{E}_{c_{pq}r} = \begin{bmatrix} 0 & 0 \end{bmatrix} \ \forall r \neq p, q, \ \mathbf{F}_{c_{pq}p} = \begin{bmatrix} 0 \end{bmatrix} \ \forall p.$$

The horizon was set to $N = 10$ and the candidate controllers were a simple constant LQR design

$$\forall p \in \{1, \ldots, 5\}: \ \mathbf{K}_p(j) = [-0.6667 \ -1.3333]$$
$$\forall j \in \{0, \ldots, N\}.$$

The algorithm and simulation were implemented in Matlab, with the optimization solved using the "linprog" linear program solver within the Matlab Optimization Toolbox. For simplicity, all of the sub-problems were solved on the same computer, running Windows on a 3.2 GHz processor with 1 GB of RAM. This is possible because the sub-problems are solved sequentially. While this is not strictly a distributed implementation, the results still illustrate the effect on computation of dividing a single optimization into a sequence of sub-problems.

For the first set of simulations, constant step disturbances were introduced at the second step

$$\mathbf{w}_p(k) = \begin{cases} \mathbf{0}, & k \leq 1 \\ \bar{\mathbf{w}}_p, & \text{otherwise} \end{cases}$$

and each subsystem was subjected to a step taken from a different vertex of the disturbance set, apart from one subsystem, which received no disturbance

$$\bar{\mathbf{w}}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \ \bar{\mathbf{w}}_2 = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}, \ \bar{\mathbf{w}}_3 = \begin{pmatrix} 0.1 \\ -0.1 \end{pmatrix},$$
$$\bar{\mathbf{w}}_4 = \begin{pmatrix} -0.1 \\ 0.1 \end{pmatrix}, \ \bar{\mathbf{w}}_5 = \begin{pmatrix} -0.1 \\ -0.1 \end{pmatrix}.$$

Figure 2 shows the results from the application of DMPC as described in §4 to this system. The state space trajectories in figure 2(a) show that the local constraints were always satisfied and each subsystem position converged to the target set. Figure 2(b) shows the time histories of the position of each mass. The shaded tube is constructed to enclose ±0.3 about the median position. Therefore, since the positions all remain within this tube, no two masses were ever more than 0.6 apart and so the coupling constraints were satisfied. Note that in both figures, the system can be seen to run right up to the constraint boundaries. Hence the key advantage that MPC is explicitly aware of constraints is retained by DMPC.

For comparison with the working example in figure 2(b), figure 3 shows position time histories for two simulations using two different controllers, each of which included a deliberate error in the implementation

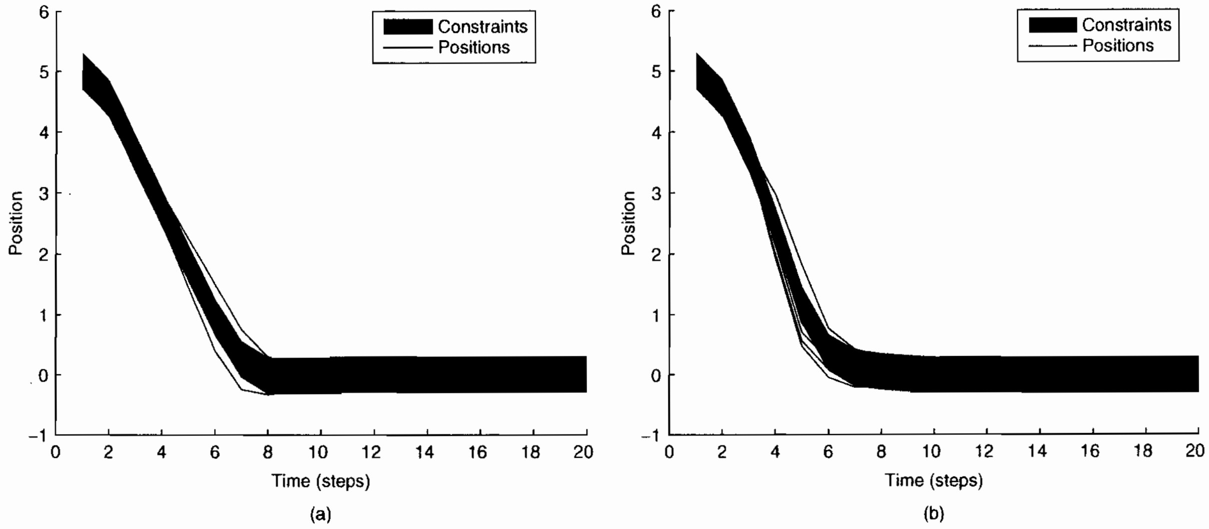


Figure 3. Simulation position time histories for comparison DMPC examples with implementation errors. Constraint violations can be seen in both cases. (a) Coupling constraints omitted; (b) Disturbance model reduced by 10%.

of DMPC. Figure 3(a) shows the results using DMPC but with the coupling constraints omitted in the optimizations. This is equivalent to using MPC independently for each subsystem subject only to its local constraints. Since the coupling constraints were violated in this case, we infer that the satisfaction of the constraints seen in figure 2(b) is a direct result of the communication in the DMPC algorithm.

Figure 3(b) shows another set of position results, this time using DMPC with coupled constraints but with the disturbance models $\mathcal{W}_p$ used for constraint tightening reduced by 10%. The disturbances applied in the simulation remained as described above, and therefore the requirement that $\mathbf{w}_p(k) \in \mathcal{W}_p$ from (2) is not satisfied. Since the guarantee of feasibility does not hold in this case, the controller was configured to apply $\mathbf{u}_p(k) = \mathbf{K}_p(0)\mathbf{x}_p(k)$ if the sub-problem $\mathcal{P}_p(\mathbf{x}_p(k))$ could not be solved. Feasibility was lost during the simulation, and the figure shows that constraint violations occurred. Therefore, since the controller fails when the disturbance exceeds its assumed bound by a relatively small amount, the success of the results shown in figure 2 can be attributed to the correct constraint tightening.

To investigate the computation and performance trades of DMPC in comparison with centralized MPC, the point mass problem was solved with both methods for varying numbers of subsystems between $N_p = 2$ and $N_p = 20$. DMPC was implemented as described in §4 and the disturbances were chosen randomly at each step from within the assumed bounds. Figure 4 compares the solution times, taken as the total of all the optimization times over a 20 step simulation, for MPC and DMPC. The solution times for DMPC are broken down per sub-problem, but the overall height of the bar is the appropriate measure as sub-problems must be solved in sequence. Note that the vertical scales are different: for $N_p = 20$ subsystems, DMPC is over five times faster than normal MPC. Figure 5 compares the performance of the two controllers, taken as the sum of the performance objective $\sum_{p=1}^{N_p}\sum_{k=1}^{20} d(\mathbf{s}_p(k), \mathcal{S}_p)$, for the same range of problems. Each bar is broken down by subsystem in both cases. The scales of both plots are identical, and hence it is clear that there is very little performance difference. Closer inspection reveals that DMPC has slightly poorer performance than MPC. This is to be expected, as DMPC approximates the overall optimization using the sub-problem sequence and therefore is a more constrained solution approach. However, taking figures 4 and 5 together, DMPC is shown to offer significant computational benefits for only a slight penalty in performance.

8. Conclusion

A formulation for distributed model predictive control (DMPC) has been presented. It solves the problem of control of multiple subsystems subjected to coupled constraints but otherwise independent. Each subsystem solves a sub-problem involving only its own state predictions. The scheme is proven to guarantee robust constraint satisfaction and convergence under the assumption of bounded disturbances. Each sub-problem is guaranteed to be feasible, convergence is guaranteed,

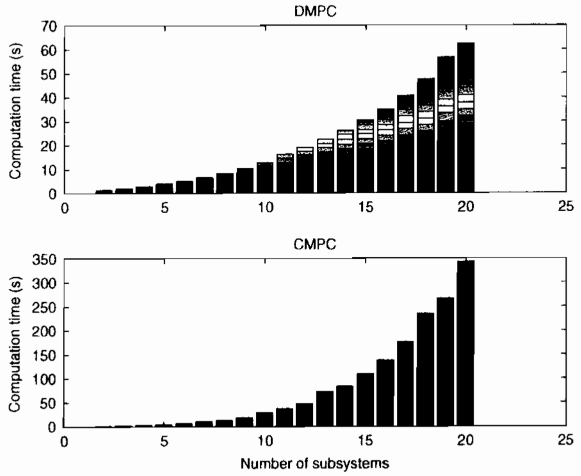


Figure 4. Solution time comparisons of MPC and DMPC for different numbers of subsystems. Note different vertical scales.

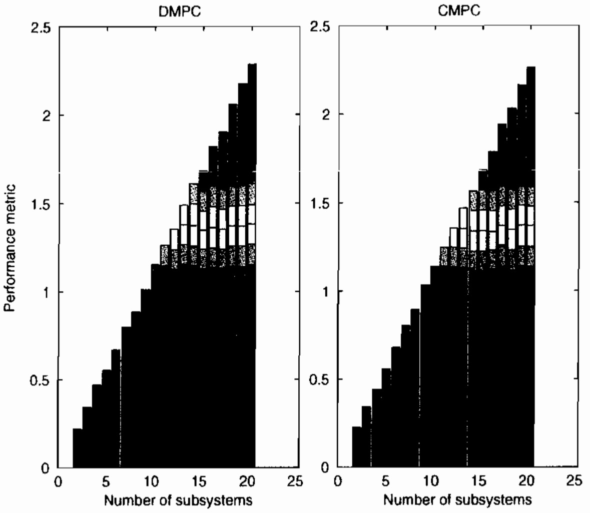


Figure 5. Performance comparisons of MPC and DMPC for different numbers of subsystems, broken down by individual subsystem.

and no iteration between subsystems is required. Results for a simple example have shown that the computation for DMPC is faster and more scalable than the equivalent centralized controller and that there is very little performance degradation in switching from centralized MPC to DMPC.

References

A. Bemporad and M. Morari, "Control of Systems integrating logic dynamics and constraints", *Automatica*, 35, pp. 407–427, 1999.

E. Camponogara, D. Jia, B.H. Krogh and S. Talukdar, "Distributed model predictive control", *IEEE Contr. Syst. Mag.*, 2, pp. 44–52, 2002.

L. Chisci, J.A. Rossiter and G. Zappa, "Systems with persistent disturbances: predictive control with restrictive constraints", *Automatica*, 37, pp. 1019–1028, 2001.

W.B. Dunbar and R.M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization", *Automatica*, 42, pp. 549–558, 2006.

M. Flint, M. Polycarpou and E. Fernandez-Gaucherand, "Cooperative path-planning for autonomous vehicles using dynamics programming", in *Proceedings of 15th IFAC World Congress*, Barcelona, 2002, pp. 1694–1699.

G. Inalhan, J.P. How and M. Tillerson, "Co-ordination and control of distributed spacecraft systems using convex optimization techniques", *Int. J. Robust and Nonlinear Contr.*, 12, pp. 207–242, 2002.

D. Jia and B. Krogh, "Min-max feedback model predictive control for distributed control with communication", in *Proceedings of American Control Conference*, Anchorage, Alaska, 2002, pp. 4507–4545.

E.C. Kerrigan, "Invariant set toolbox for matlab", Available online at http://www-control.eng.cam.ac.uk/eck21/matlab/invsetbox/index.html (accessed 20 June 2005).

E.C. Kerrigan and J.M. Maciejowski, "Robust feasibility in model predictive control", in *Proceedings of 40th IEEE Conference on Decision and Control*, Orlando, Florida, 2001, p. 728.

E.C. Kerrigan and J.M. Maciejowski, "Feedback min-max model predictive control using a single linear program: robust stability and the explicit solution", *Int. J. Robust and Nonlinear Contr.*, 14, pp. 395–413, 2004.

T. Keviczky, F. Borrelli and G.J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems", *Automatica*, 42, pp. 2105–2115, 2006.

I. Kolmanovsky and E. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems", *Math. Prob. Eng.: Theory, Methods and Appl.*, 4, pp. 317–367, 1998.

Y. Kuwata, A. Richards, T. Schouwenaars and J.P. How, "Decentralized robust receding horizon control for multi-vehicle guidance", in *Proceedings of the American Control Conference*, Minneapolis, Minnesota, 2006, pp. 2047–2052.

K.V. Ling, J.M. Maciejowski and B.F. Wu, "Multiplexed model predictive control", in *Proceedings of 16th IFAC World Congress*, Prague, 2005.

J.M. Maciejowski, *Predictive Control with Constraints*, Pearson Education Ltd., Harlow, UK: Prentice Hall, 2002.

L. Magni and R. Scattolini, "Stabilizing decentralized model predictive control of nonlinear systems", *Automatica*, 42, pp. 1231–1236, 2006.

D.Q. Mayne, J.B. Rawlings, C.V. Rao and P.O.M. Scokaert, "Constrained model predictive control: stability and optimality", *Automatica*, 36, pp. 789–814, 2000.

M. Pachter and P.R. Chandler, "Challenges of autonomous control", *IEEE Contr. Syst. Mag.*, 18, pp. 9297, 1998.

S.J. Qin and T.A. Badgwell, "An overview of industrial model predictive control technology", in *Proceedings of Fifth International Conference on Chemical Process Control*, J. Kantor, C. Garcia and B. Carnahan, Eds, Tahoe City, California, Published by AICHE, New York, 1997, pp. 232–256.

R.L. Raffard, C.J. Tomlin and S.P. Boyd, "Distributed optimization for cooperative agents: application to formation flight", in *Proceedings of the 43rd IEEE Conference on Decision and Control*, 3, 2004, pp. 2453–2459.

C.V. Rao, S.J. Wright and J.B. Rawlings, "Application of interior-point methods to model predictive control", *J. Opt. Theory and Appl.*, 99, pp. 723–757, 1998.

A.G. Richards and J.P. How, "Decentralized model predictive control of cooperating UAVs", in *Proceedings of IEEE Conference on Decision and Control*, Paradise Island, Bahamas, 2004, pp. 4286–4289.

A.G. Richards and J.P. How, "Implementation of robust decentralized model predictive control", in *AIAA Guidance, Navigation and Control Conference*, San Francisco, CA, 2005, AIAA-2005-6366.

A.G. Richards and J.P. How, "Robust stable model predictive control with constraint tightening", in *Proceedings of American Control Conference*, Minneapolis, Minnesota, 2006, pp. 1557–62.

T. Schouwenaars, J.P. How and E. Feron, "Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees", in *Proceedings of AIAA Guidance Navigation and Control Conference*, Providence, Rhode Island, 2004, AIAA-2004-5141.

P.O.M. Scokaert and D.Q. Mayne, "Min-max model predictive control for constrained linear systems", *IEEE Trans. Automat. Contr.*, 43, pp. 1136–1142, 1998.

D.H. Shim, H.J. Kim and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots in dynamic environment", in *Proceedings of IEEE Conference on Decision and Control*, Maui, Hawaii, 2003, pp. 3621–3626.

M.G. Singh and A. Titli, *Systems: Decomposition, Optimisation and Control*, New York: Pergamon Press, 1978.

A.N. Venkat, J.B. Rawlings and S.J. Wright, "Plant-wide optimal control with decentralized MPC", Technical report, Department of Chemical and Biological Engineering, University of Wisconsin (2004).

S.L. Waslander, G. Inalhan and C.J. Tomlin, "Decentralized optimization via Nash bargaining", in *Conference on Cooperative Control and Optimization*, Destin, Florida, 2003, pp. 565–583.

J. Yan and R.R. Bitmead, "Incorporating state estimation into model predictive control and its application to network traffic control", *Automatica*, 41, pp. 595–604, 2005.

Cooperative Decentralized Robust Trajectory Optimization using Receding Horizon MILP

Yoshiaki Kuwata *Member, IEEE,* and Jonathan P. How *Senior Member, IEEE*

Abstract

Motivated by recent research on cooperative UAVs, this paper introduces a new decentralized trajectory optimization approach for systems with independent dynamics but coupled constraints and objectives. The overall goal is to develop a decentralized approach that solves small subproblems while minimizing a fleet-level objective. In the new algorithm, vehicles solve their subproblems in sequence, while generating feasible modifications to the prediction of other vehicles' plans. In order to avoid reproducing the global optimization, the decisions of other vehicles are parameterized using a much smaller number of variables than in the centralized formulation. This reduced number of variables is sufficient to improve the cooperation between vehicles without significantly increasing the computational effort involved. The resulting algorithm is shown to be robustly feasible under the action of unknown but bounded disturbances. Furthermore, the fleet objective function is proven to monotonically decrease while going through the vehicles in the fleet and over the time. Simulation results are presented to compare the distributed, centralized, and other (non-cooperative) decentralized approaches in terms of both computation and performance. The results from simulations and a hardware experiment demonstrate that the proposed algorithm can improve the fleet objective by temporarily having one vehicle sacrifice its individual objective, showing the cooperative behavior.

Index Terms

Cooperative control, Distributed, Multi-vehicle experiments, Robust Model Predictive Control (MPC).

I. Introduction

Teams of multiple UAVs have many applications, such as border patrol, cooperative search and track, and mobile sensor network [1]–[5]. To enable fleet-level cooperation, the overall control system must properly capture the complex interactions between the vehicles and tasks. One approach is to solve this problem globally, but centralized algorithms typically scale very poorly with the fleet size because of the computational effort involved. A natural approach to decompose the centralized problem is to let each vehicle optimize its own decision variables. A key challenge of decentralized control is to ensure that the distributed decision making leads to actions that satisfy the coupling constraints with other vehicles [6], [7], so that the fleet as a whole executes consistent plans. Various approaches have been investigated to address this problem, including treating the influence of other subsystems as an unknown disturbance [8], coupling penalty functions [9]–[11], partial grouping of computations [12], loitering options for safety guarantees [13], and dynamic programming [14], [15]. Some approaches involve iterative negotiations between subsystems [11], [16] and apply game theory to study convergence. Decentralization is further complicated when disturbances act on the subsystems, making the prediction of the future behavior uncertain.

Research funded by AFOSR FA9550-04-1-0458 and DURIP F49620-02-1-0216.

Y. Kuwata, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA 02139, USA, kuwata@alum.mit.edu

J. How, Professor, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA 02139, USA, jhow@mit.edu

Manuscript received March 30, 2007 .

For the trajectory planning problem, several distributed control architectures have been proposed that design paths locally from the current vehicle location. Much of the current research on the distributed path planning assumes that each vehicle solves a local problem and communicates this intent information to its neighbors [13], [17]–[21]. These decentralized algorithms typically lead to a Nash equilibrium [22] or a Pareto optimal surface [17], [23], which is not necessarily the globally optimal solution, because these so-called "communication-based approaches" [24] do not use the information about the objective functions of the other subsystems and do not consider the overall performance. For example, the recently proposed trajectory planning algorithm called *Decentralized Robust Safe but Knowledgeable* (DRSBK) generates a local plan over a short horizon while guaranteeing the robust feasibility of the entire fleet under the action of external disturbances [20]. This algorithm uses distributed robust Model Predictive Control [18] to predict and account for other vehicles' behavior. However, because each vehicle only optimizes for its own control and freezes the decisions of other vehicles, the resulting trajectories can be locally optimal but globally suboptimal (i.e., coordinated but *non-cooperative*) [24].

This paper extends the DRSBK algorithm to enable *cooperation* amongst the vehicles in the fleet. Each vehicle solves its subproblem in sequence to optimize its own control input, as in DRSBK, but the subproblems can also explicitly perturb the decisions of other vehicles to improve the global cost. The challenge here is to avoid reproducing the global optimization for each vehicle. The proposed approach uses a low-order parametrization of the other vehicles' decisions to reduce the solution space while retaining the freedom to alter key aspects of other vehicles' plans. This effectively enables a negotiation between the vehicles with only a small increase in the computational complexity of the subproblem optimizations.

The paper begins with the problem setup in Section II. Following the centralized algorithm in Section III and the decentralized non-cooperative algorithm in Section IV, Section V presents the cooperative form of robust decentralized trajectory optimization algorithm using receding horizon MILP. Section VI proves that the new algorithm retains the robust feasibility and that each solution of the subproblem monotonically decreases the global cost. Section VII shows simulation results and compares the performance and computation time of the algorithm. Finally, experimental results are presented in Section VIII.

II. Problem Statement

Notation: In this paper, the index or superscript p, q denotes the vehicle index, index k denotes the current time step, and index j denotes the prediction step. There are total of n vehicles. Unless otherwise noted, $\forall p$ implies $\forall p = 1, \dots, n$, and $\forall q$ implies $\forall q = 1, \dots, n$ but $q \neq p$. The neighbor set $\mathcal{N}^p_k$ is a set of vehicles that have coupling constraints with vehicle p at time k. It also determines the order that the vehicles solve their subproblems. $\mathtt{pre}(p)$ and $\mathtt{next}(p)$ respectively denote the vehicle that solves the subproblem immediately before and after vehicle p.

A fleet of n vehicles are assumed to have independent dynamics, which are described by the LTI model

$$\boldsymbol{x}^p_{k+1} = A\boldsymbol{x}^p_k + B\boldsymbol{u}^p_k + \boldsymbol{w}^p_k, \quad \forall p, \forall k \tag{1}$$

where $\boldsymbol{x}^p_k$ is the state vector of size n_x, $\boldsymbol{u}^p_k$ is the input vector of size n_u, and $\boldsymbol{w}^p_k$ is the disturbance vector for the p^{th} vehicle. The disturbances $\boldsymbol{w}^p_k$ are unknown but lie in known bounded sets $\boldsymbol{w}^p_k \in \mathcal{W}^p$. The environment has obstacles to be avoided, and the vehicles have flight envelope limitations. The general output sets $\mathcal{Y}^p$ capture these local constraints of each vehicle

$$\boldsymbol{y}^p_k = C\boldsymbol{x}^p_k + D\boldsymbol{u}^p_k \in \mathcal{Y}^p, \quad \forall p, \forall k. \tag{2}$$

The coupling between vehicles, such as vehicle avoidance, inter-vehicle communication range, and line-of-sight between vehicles is captured by a further set of constraints applied to the sum of outputs from each vehicle

$$\boldsymbol{z}^p_k = E^p\boldsymbol{x}^p_k, e \quad \forall p, \forall k \tag{3a}$$

$$\sum_{p=1}^{n} \boldsymbol{z}^p_k \in \mathcal{Z}. \tag{3b}$$

For pair-wise collision avoidance constraints, there are only two matrices E^p and E^q that have nonzero entries for a given row of the output matrix E, and the set $\mathcal{Z}$ is non-convex.

Finally, the goal of the trajectory optimization is to minimize the fleet cost composed of individual costs, which could be conflicting. Optimization is performed to obtain the optimal input sequence for each vehicle

$$\min_{\boldsymbol{u}_k^p, \forall p, \forall k} J\left(J^1, \ldots, J^n\right). \tag{4}$$

III. Centralized Approach

The centralized approach directly solves the full optimization in Section II. Let U^p denote the control input sequence of the vehicle p over the next N steps, i.e., $U^p = \left[\boldsymbol{u}_0^{p\mathrm{T}}, \ldots, \boldsymbol{u}_{N-1}^{p\;\mathrm{T}}\right]^{\mathrm{T}}$. Then, a compact form of the optimization can be written as

$$\begin{aligned} &\min_{U^1,\ldots,U^n} \quad J\left(J^1, \ldots, J^n\right) \\ &\text{subject to} \quad \forall p: \quad g(U^p) \leq 0 \\ &\qquad\qquad\qquad\qquad h(U^p, U^q) \leq 0, \quad q \in \mathcal{N}^p \end{aligned} \tag{5}$$

where $g(U^p)$ represents the local constraints for vehicle p, and $h(U^p, U^q)$ represents the coupling constraints between vehicles p and q. This approach produces the globally optimal solution; however, it scales poorly because the optimization becomes very complex for large fleets for most problem types (i.e., quadratic programming and mixed-integer linear programming) [25].

IV. Decentralized Robust RHC

This section briefly describes the DRSBK algorithm presented in [20], which is a decentralized approach that decomposes the centralized problem (5) into smaller subproblems. DRSBK is a receding horizon algorithm, which solves online optimization that develops the control inputs for a finite horizon of N steps, and implements only the first step. The optimization is repeated as the system evolves. To reduce the computational burden, DRSBK uses a short planning horizon, and the remainder of the path to the target is represented by a sophisticated cost-to-go function that is cognizant of obstacles in the cluttered environment. Invariance constraints are imposed at the terminal step of the short plan to ensure the safety of the vehicle against the potential changes in the environment beyond the planning horizon. Each vehicle generates its control inputs by solving a subproblem in sequence, while freezing the plan of other vehicles. The solution is then communicated to other vehicles in the fleet. The advantages of this algorithm include robust constraint satisfaction under the action of disturbance, much better scalability compared to the centralized approach, and requiring only local communication [20].

To simplify the presentation, the planning order is assumed to be $1, \ldots, n$. The prediction of a value at time $(k+j)$ made at time k is denoted by subscript $(\cdot)_{k+j|k}$. Unless otherwise noted, $\forall j$ implies $j = 0, \ldots, N-1$, and $\forall j^-$ implies $\forall j = 0, \ldots, N-2$. At time k, the p^{th} subsystem generates its own control inputs $\boldsymbol{u}_{\cdot|k}^p$ by solving the following subproblem

$$J^{p*} = \min_{\boldsymbol{u}_{\cdot|k}^p} f^p\left(\boldsymbol{x}_{k+N|k}^p\right) \tag{6}$$

$$\text{s.t. } \forall j: \qquad \boldsymbol{x}_{k|k}^p = \boldsymbol{x}_k^p \tag{7}$$

$$\boldsymbol{x}_{k+j+1|k}^p = A^p \boldsymbol{x}_{k+j|k}^p + B^p \boldsymbol{u}_{k+j|k}^p \tag{8}$$

$$\boldsymbol{y}_{k+j|k}^p = C^p \boldsymbol{x}_{k+j|k}^p + D^p \boldsymbol{u}_{k+j|k}^p \in \mathcal{Y}_j^p \tag{9}$$

$$\boldsymbol{z}_{k+j|k}^p = E^p \boldsymbol{x}_{k+j|k}^p \tag{10}$$

$$\boldsymbol{z}_{k+j|k}^p + \tilde{\boldsymbol{z}}_{k+j|k}^p \in \mathcal{Z}_j^p \tag{11}$$

$$\boldsymbol{x}_{k+N|k}^p \in \mathcal{Q}_k^p. \tag{12}$$

Here, the objective function for the entire fleet is assumed to be the sum of the individual costs. The vector $\tilde{\boldsymbol{z}}_{k+j|k}^p$ is a summation of the outputs from the other vehicles

$$\tilde{\boldsymbol{z}}_{k+j|k}^p = \sum_{\substack{q \in \mathcal{N}_k^p, \\ q < p}} \boldsymbol{z}_{k+j|k}^q + \sum_{\substack{q \in \mathcal{N}_k^p, \\ q > p}} \boldsymbol{z}_{k+j|k-1}^q \tag{13}$$

and is constant in this local optimization. The first summation is from the vehicles that have already planned at time k. The second summation is from the vehicles that have not planned at time k, so that a prediction made at the previous time $(k-1)$ is used.

The prediction (8) does not include the disturbance, but the constraint sets $\mathcal{Y}_j^p$ differ from the original set $\mathcal{Y}^p$. To save some margin to account for the uncertainties in the future, the set $\mathcal{Y}^p$ is tightened with the prediction step j to form $\mathcal{Y}_j^p$. A linear controller is used to rejects the disturbance [26]

$$\begin{aligned} &\mathcal{Y}_0^p = \mathcal{Y}^p \\ &\mathcal{Y}_{j+1}^p = \mathcal{Y}_j^p \sim (C^p L_j^p + D^p P_j^p)\mathcal{W}^p \quad \forall j^- \end{aligned} \tag{14}$$

where the operator $\sim$ denotes the Pontryagin difference [27], and L_j^p is the state transition matrix

$$\begin{aligned} &L_0^p = I \\ &L_{j+1}^p = A^p L_j^p + B^p P_j^p \quad \forall j^- . \end{aligned} \tag{15}$$

Similar tightening is performed on the coupling constraint sets in (11), allowing uncertainty margin for all subsystems

$$\mathcal{Z}_0^{p_n} = \mathcal{Z} \tag{16a}$$

$$\mathcal{Z}_j^{\mathtt{pre}(q)} = \mathcal{Z}_j^q \sim E^q L_j^q \mathcal{W}^q, \quad \forall j, \quad q \in \mathcal{N}_k^p, \ q \neq p_0 \tag{16b}$$

$$\mathcal{Z}_{j+1}^{p_n} = \mathcal{Z}_j^{p_0} \sim E^{p_0} L_j^{p_0} \mathcal{W}^{p_0}, \quad \forall j^- \tag{16c}$$

where p_0 and p_n represent the first and the last element of the neighbor set $\mathcal{N}_k^p$, respectively. The operation (16b) tightens the constraints from the vehicle q to $\mathtt{pre}(q)$. This tightening represents that the vehicle $\mathtt{pre}(q)$, which generates the plan prior to vehicle q, saving some margin for the vehicle q so that q can use it to reject the disturbances $\mathcal{W}^q$. The operation (16c) tightens the constraints from the prediction step j to $j+1$. This represents that the optimization at time k for vehicle p_n saves some margin so that the optimization at time $(k+1)$ for vehicle p_0 can use it to also reject the disturbances.

The set $\mathcal{Q}_k^p$ in (12) is called a *safety* set, and the terminal states of the plan must end in this set [28]. By using nilpotent candidate controllers P_j^p, which makes $L_{N-1}^p = 0$ in (15), the safety set can be expressed as a nominal control invariant admissible set, which has the following invariance property

$$\begin{aligned} &\forall \boldsymbol{x}^p \in \mathcal{Q}_k^p \Rightarrow \exists \kappa^p(\boldsymbol{x}^p) \ \text{ s.t.} \\ &\qquad A^p \boldsymbol{x}^p + B^p \kappa^p(\boldsymbol{x}^p) \in \mathcal{Q}_k^p \\ &\qquad C^p \boldsymbol{x}^p + D^p \kappa^p(\boldsymbol{x}^p) \in \mathcal{Y}_{N-1}^p \\ &\qquad E^p \boldsymbol{x}^p + \sum_{q \in \mathcal{N}_k^p} E^q \boldsymbol{x}^q \in \mathcal{Z}_{N-1}, \qquad \forall (\boldsymbol{x}^{p_0}, \ldots, \boldsymbol{x}^{p_n}) \in \{\mathcal{Q}_k^{p_0} \times \ldots \times \mathcal{Q}_k^{p_n}\}. \end{aligned} \tag{17}$$

where for the vehicles that have not planned at time k, the invariant set obtained in the previous time $(k-1)$ is used, i.e.,

$$\mathcal{Q}_k^q = \mathcal{Q}_{k-1}^q, \ \forall q \in \{\mathtt{next}(p), \ldots, p_n\}$$

and $\kappa^p(\boldsymbol{x}^p)$ is a nonlinear control law that the vehicle can use, once in $\mathcal{Q}_k^p$, to stay in $\mathcal{Q}_k^p$. For fixed-wing aircraft, one invariance example is a loiter circle where $\kappa^p(\boldsymbol{x}^p)$ generates a centripetal acceleration perpendicular to the heading direction. For rotorcraft, any point with zero velocity is invariant with $\kappa^p(\boldsymbol{x}^p) = 0$.

DRSBK minimizes the terminal state penalty $f^p(\boldsymbol{x}_{k+N|k}^p)$ as the objective function in (6), but the objective function can readily include the stage cost such as control penalty. For the environment with no-fly zones, a sophisticated cost-to-go function is used as the terminal penalty that is calculated in two stages [29]. First, prior to online optimization, shortest path algorithm is applied to a graph-based representation of the environment to estimate the approximate cost $\tilde{f}^p(\boldsymbol{r}_{\text{corner}})$ to fly from each obstacle corner $\boldsymbol{r}_{\text{corner}}$ to the target. The pairs of corner locations $\boldsymbol{r}_{\text{corner}}$ and the cost $\tilde{f}^p(\boldsymbol{r}_{\text{corner}})$ are stored as a cost map. Then, the optimization chooses the best corner $\boldsymbol{r}_{\text{vis}}^p$ that is visible from a point $\boldsymbol{O}^p$ in the invariant safety set $\mathcal{Q}_k^p$ of the vehicle p, so that the cost-to-go function is

$$J^p = f^p(\boldsymbol{x}_{k+N|k}^p) = \|\boldsymbol{O}^p - \boldsymbol{r}_{\text{vis}}^p\|_2 + \tilde{f}^p(\boldsymbol{r}_{\text{vis}}^p). \tag{18}$$

For notational simplicity in the later sections, let $\mathsf{C}^p(\boldsymbol{x}_k^p, U_k^p)$ denote the set of constraints (7)–(12), (18) for vehicle p. The first argument $\boldsymbol{x}_k^p$ is the initial condition used in the right hand side of (7), and the second argument U_k^p denotes the control input in the optimization, i.e., $U_k^p = \left[\boldsymbol{u}_{k|k}^{p\ \mathsf{T}}, \ldots, \boldsymbol{u}_{k+N-1|k}^{p\ \mathsf{T}}\right]^{\mathsf{T}}$.

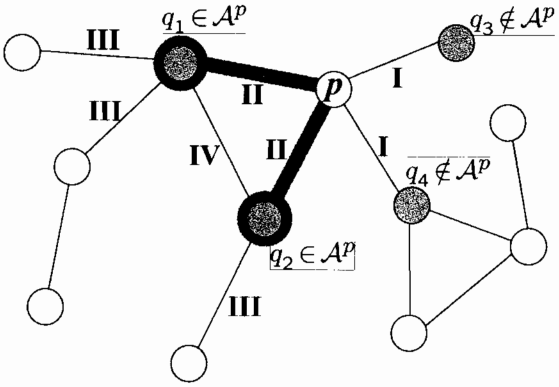


Fig. 1. Node p's neighbor set $\mathcal{N}$ and active coupling neighbor set $\mathcal{A}$, and different types of coupling constraints.

The optimization (6)–(12), (18) is implemented using mixed-integer linear programming (MILP), which can express logical decisions such as the choice of r^p_{vis}, and non-convex constraints such as collision avoidance. From the solution at time k, the first control input $u^p_{k|k}$ for each vehicle is applied to the real system (1). At the next time $k+1$, the states of each vehicle x^p_{k+1} are measured, and the optimization is repeated. The main result of DRSBK presented in [20] is the robust feasibility guarantee, i.e. feasibility of the optimization (7)–(12) at time k implies feasibility at time $k+1$ under the action of disturbance $\forall w^p_k \in \mathcal{W}^p$ [30].

The primary advantage of this approach is that the decision space of the subproblem is approximately n times smaller than the centralized approach, with many fewer constraints. As a result, the computation time is much smaller, and the algorithm scales much better than the centralized approach. Furthermore, all the constraints are robustly satisfied for all vehicles at each time step. However, since each vehicle does not account for the objectives of other vehicles, the resulting solution is *coordinated* but *non-cooperative*. This non-cooperative solution is a Nash equilibrium, where no vehicle can improve its local cost by changing only its own decision, and is undesirable. The benefits and limitations of this approach are clearly illustrated in the examples in Section VII-B.

V. Cooperative DRSBK Algorithm

This section extends the DRSBK algorithm [20] to enable explicit cooperation. The DRSBK algorithm guarantees the robust constraint satisfaction under the action of disturbances. However, because the objective function (6) does not consider the effect from/on the other vehicles as in (4), the resulting solution could be a Nash equilibrium, where the solution is locally optimal but globally suboptimal. One intuitive approach to resolve this issue is to include all the decision variables of other vehicles in each subproblem. However, this will reproduce the global optimization for each vehicle and is clearly not scalable. The proposed approach exploits the sparse structure of the coupling constraints of trajectory optimization and uses the low-order parametrization of the other vehicles to reduce the dimension of the decision space.

A. Algorithm Overview

This subsection gives a brief overview of the algorithm to present the main idea. Whereas DRSBK freezes all the decisions of other vehicles, the cooperative form of DRSBK (called CDRSBK) updates the candidate solutions of other vehicles by designing a feasible perturbation to other vehicles' decisions. Each vehicle sequentially solves its own subproblem as well as slightly modified subproblems of other vehicles. The algorithm iterates over the team of vehicles, but the simulation results in Section VII-B show that two iterations over the fleet typically produces a performance comparable to the centralized approach.

Let $\mathcal{A}^p$ denote a set of vehicles that have active coupling constraints with vehicle p. Figure 1 shows an example of a graphical representation of a vehicle fleet. Each node represents a vehicle, and the arc connecting two nodes shows that there is a coupling constraint between the two vehicles. The shaded nodes in the figure are the neighbors of vehicle p, i.e., $q_1, \dots, q_4 \in \mathcal{N}^p$. In general, not all of the coupling constraints are active. In this example, active coupling neighbors of vehicle p are marked with thick lines, and $q_1, q_2 \in \mathcal{A}^p$ but $q_3, q_4 \notin \mathcal{A}^p$.

The centralized optimization (5) is broken into the following local optimization for each vehicle p:

$$\min_{\substack{U^p \\ \xi^q,\, q \in \mathcal{A}^p}} \ J \tag{19}$$

subject to

$$
\begin{aligned}
g(U^p) &\le 0 && && \text{(20a)}\\
g(\bar{U}^q + T^q\xi^q) &\le 0, && q \in \mathcal{A}^p && \text{(20b)}\\
h(U^p, \bar{U}^q) &\le 0, && q \in \mathcal{N}^p,\ q \notin \mathcal{A}^p && \text{(20c)}\\
h(U^p, \bar{U}^q + T^q\xi^q) &\le 0, && q \in \mathcal{A}^p && \text{(20d)}\\
h(\bar{U}^r, \bar{U}^q + T^q\xi^q) &\le 0, && r \in \mathcal{N}^q, r \notin \mathcal{A}^p, q \in \mathcal{A}^p && \text{(20e)}\\
h(\bar{U}^{q_1} + T^{q_1}\xi^{q_1}, \bar{U}^{q_2} + T^{q_2}\xi^{q_2}) &\le 0, && q_1, q_2 \in \mathcal{A}^p. && \text{(20f)}
\end{aligned}
$$

The main difference from the DRSBK algorithm is that the decision variables include U^p for vehicle p itself and ξ^q for other neighboring vehicles with active couplings. The decision for q made in p's optimization is a perturbation from the communicated solution $\bar{U}^q$, so that the control input of other vehicle q is written as

$$U^q = \bar{U}^q + \delta U^q = \bar{U}^q + T^q\xi^q, \quad \forall q.$$

The perturbation δU^q is rewritten as $\delta U^q = T^q\xi^q$ using a parametrization matrix T^q, which is elaborated later.

The optimization includes the constraints for vehicles p and $\forall q \in \mathcal{A}^p$. The constraints (20a)–(20b) are the local constraints for vehicle p and for its active coupling neighbors. The equations (20c)–(20f) express different types of couplings shown in Figure 1. Type I is between vehicle p and its neighbors with no active couplings; Type II is between vehicle p and its neighbors with active couplings; Type III is between vehicle p's active coupling neighbors and their neighbors; Type IV is between vehicle p's two active coupling neighbors. Note that some constraints in $h(U^p, \bar{U}^q + T^q\xi^q) \le 0$ could be omitted if ξ^q has no impact on them because of the row rank deficiency of T^q, which is discussed in more detail in Section V-B.

A key advantage of this algorithm is that it does not freeze the other vehicles' plans, so it can avoid the Nash equilibrium obtained from other algorithms. The next section gives a method for computing the parametrization matrix T^q that reduces the computational complexity of each local optimization.

B. Reduced-order Decision Space

This subsection discusses the approach for reducing the decision space and how to compute the corresponding parametrization matrix T^q.

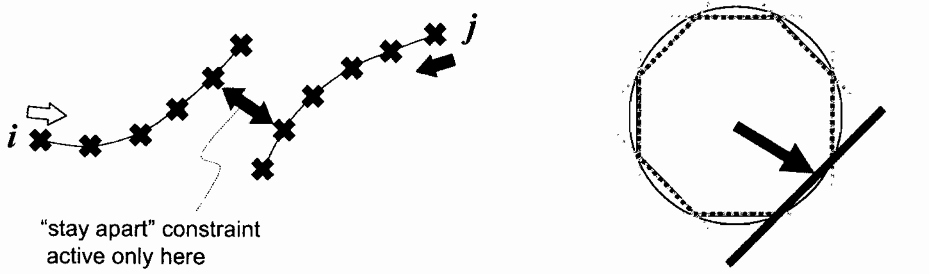
1) Active Coupling Constraints: In a typical trajectory optimization, not all of the coupling constraints are active in the future plans. For example, collision avoidance problems typically have one or two time steps when the vehicles are close, as shown in Figure 2(a). Furthermore, the relative distance, i.e., the 2-norm of the relative position vector, can be expressed with a set of linear constraints, but only one of these can be active, as shown in Figure 2(b). The approach presented here exploits this sparse structure of the active coupling constraints to reduce the decision space. In particular, a perturbation is only generated when there exists an active coupling constraint between the vehicles.

It is assumed that the coupling constraints take a linear form

$$F^pU^p + F^qU^q \le g^{pq} \tag{21}$$

which can express convex constraints including 1-norm, 2-norm, and ∞-norm bounds. Using binary variables, non-convex constraints can be also expressed in this form in MILP. The subproblem of vehicle p in DRSBK fixed the decision of other vehicles so that (21) is

$$F^pU^p + F^q\bar{U}^q \le g^{pq}.$$



(a) Trajectory of two vehicles. The collision avoidance constraint is active only at one time step.

(b) Two-norm constraints expressed with a set of linear constraints.

Fig. 2. "Sparsity" examples in the trajectory optimization

One cooperative form is to change the right hand side in p's optimization

$$F^p U^p + F^q \bar{U}^q \leq g^{pq} - \beta \tag{22}$$

and implicitly expect the vehicle q will take advantage of this change in q's next optimization. The variable β only needs to change the rows corresponding to the active coupling constraints, because a change in the active coupling constraints can directly change the decision space U^q, leading to the potential change in the cost of vehicle q. The following form expresses the decision change of q more explicitly.

$$F^p U^p + F^q(\bar{U}^q + \delta U^q) \leq g^{pq} \tag{23}$$

Without loss of generality, the upper rows of (22)–(23) can be regarded as active and the lower rows as inactive. By comparing (22) and (23), we have

$$\left[\begin{array}{c} F^q_{\text{active}} \\ \hline F^q_{\text{inactive}} \end{array}\right] \delta U^q = \left[\begin{array}{c} \beta_{\text{active}} \\ \hline * \end{array}\right].$$

Now the goal is to change β_{active} by δU^q. Note that because there are only a few coupling constraints that are active, $\dim(\beta_{\text{active}})$ is small. The key here is that δU^q can be parameterized using a variable of smaller dimension.

Let m denote the row rank of F^q_{active}, which is also the number of elements in β_{active} that any δU^q can change independently. Therefore, a new variable $\xi^q \in \mathbb{R}^m$ could replace δU^q, where in the trajectory optimization problems the dimension m of ξ^q is significantly smaller than the dimension of δU^q. Let $\breve{A}$ denote a matrix composed of the m independent row vectors extracted from F^q_{active}. Then, δU^q is parameterized by ξ^q as

$$\delta U^q = \breve{A}^T \left(\breve{A}\breve{A}^T\right)^{-1} \xi^q \triangleq T^q \xi^q. \tag{24}$$

The inverse in this equation exists because the product $\left(\breve{A}\breve{A}^T\right)$ is of full rank m, so the parametrization matrix T^q also exists. The dimension of ξ^q is much smaller than that of the original control input variables δU^q_k, so the row size of T^q is larger than the column size. In the examples considered in Sections VII-C and VIII, the row size of T^q is 2–6 times larger than the column size, illustrating the sparseness assumption.

2) Terminal States: Extra degrees of freedom for the perturbation to q's decision could be obtained by parameterizing the decision variables through the terminal states in addition to the active coupling constraints. The main motivation is that the performance of DRSBK algorithm depends strongly on the location of the invariant safety set in which the terminal states $x^q_{k+N|k}$ lie. Since

$$x^q_{k+N|k} = x^q_{k|k} + [A^{N-1}B, \cdots, AB, B] \begin{bmatrix} u^q_{k|k} \\ \vdots \\ u^q_{k+N-1|k} \end{bmatrix} \triangleq x^q_{k|k} + CU^q_k,$$

the perturbation in the control inputs are parameterized as

$$\delta U_k^q = \mathcal{C}^{\mathrm{T}}(\mathcal{C}\mathcal{C}^{\mathrm{T}})^{-1}\delta \boldsymbol{x}_{k+N|k}^q \triangleq T^q \xi^q. \tag{25}$$

This parametrization matrix T^q always exists when the system is controllable and hence the matrix $\mathcal{C}$ has full row rank. The parametrization (25) reduces the dimension of the decision space from δU_k^q, which is Nn_u, to $\delta \boldsymbol{x}_{k+N|k}^q$, which is n_x. Note that for collision avoidance problem, if vehicles p and q are far apart and do not interact with each other, the variable ξ^q that generates perturbation δU_k^q has zero dimension in p's optimization because the coupling constraints are not active.

3) Constrained Terminal States: When an invariant constraint is imposed on the terminal states of the plan (e.g., the hovering constraint for rotorcraft), the parametrization matrix must be formed differently. This subsection discusses the parametrization when the terminal velocity cannot be perturbed. Let $C_{\text{pos}} \triangleq [I,\ O]$ and $C_{\text{vel}} \triangleq [O,\ I]$ for a double integrator system, where

$$\delta \boldsymbol{x}_{k+N|k}^q = \begin{bmatrix} \delta \boldsymbol{r}_{k+N|k}^q \\ \delta \boldsymbol{v}_{k+N|k}^q \end{bmatrix} = \begin{bmatrix} C_{\text{pos}} \\ C_{\text{vel}} \end{bmatrix} \mathcal{C}\, \delta U_k^q$$

In order to keep the same terminal velocity, i.e., $\delta \boldsymbol{v}_{k+N|k}^q = 0$, the perturbation δU_k^q must lie in the null space of $C_{\text{vel}}\mathcal{C}$. Such δU_k^q can be realized using a new vector η

$$\delta U_k^q = E\eta \tag{26}$$

where E denotes an orthonormal basis for the null space of $C_{\text{vel}}\mathcal{C}$. With (26), the perturbation to the terminal position is written as

$$\delta \boldsymbol{r}_{k+N|k}^q = C_{\text{pos}}\mathcal{C}E\eta.$$

By making the perturbation $\delta \boldsymbol{r}_{k+N|k}^q$ as the decision variable ξ^q, η is parameterized as

$$\eta = (C_{\text{pos}}\mathcal{C}E)^{\mathrm{T}}\big((C_{\text{pos}}\mathcal{C}E)(C_{\text{pos}}\mathcal{C}E)^{\mathrm{T}} \big)^{-1}\xi^q.$$

Combining this with (26) gives

$$\delta U_k^q = E(C_{\text{pos}}\mathcal{C}E)^{\mathrm{T}}\big((C_{\text{pos}}\mathcal{C}E)(C_{\text{pos}}\mathcal{C}E)^{\mathrm{T}} \big)^{-1}\xi^q \triangleq T^q\xi^q. \tag{27}$$

This δU_k^q changes the terminal position $\boldsymbol{r}_{k+N|k}^q$ without changing the terminal velocity $\boldsymbol{v}_{k+N|k}^q$. Note that ξ^q in (27) has the same dimension as the position, whereas the dimension of ξ^q in (25) is n_x.

4) Fixed Binary Variables: To account for non-convex constraints such as vehicle/obstacle avoidance or minimum speed, the overall optimization is implemented using MILP. Binary variables in MILP encode logical constraints or non-convex constraints but are the major source of the computational complexity in the MILP solution process. With the goal of obtaining small perturbations for other vehicles' decision, the CDRSBK algorithm fixes binaries of other vehicles, while solving for perturbations of their continuous variables.

Figure 3 shows the effect of fixed binaries used to express the obstacle avoidance constraints (detailed equations given by (40) in Appendix). In region **A**, the admissible binaries are [0, 1, 1, 1] where 1 means the avoidance constraint is relaxed. In region **C**, the binaries are [1, 1, 1, 0]. The regions $\mathbf{B}_1$, $\mathbf{B}_2$ have three possible binary settings [0, 1, 1, 1], [1, 1, 1, 0], and [0, 1, 1, 0], and the output of the MILP solver could be any of them. If the binaries are fixed [0, 1, 1, 0] for a point in region **B**, which is a union of $\mathbf{B}_1$ and $\mathbf{B}_2$, then the perturbed point must also stay inside the region **B** only. To enable larger perturbations, CDRSBK algorithm performs a MILP pre-processing and sets [0, 1, 1, 1] as the binaries of the point in $\mathbf{B}_1$, which allows the point to move within **A**, $\mathbf{B}_1$, and $\mathbf{B}_2$. Similarly, the point in $\mathbf{B}_2$ uses the binary setting [1, 1, 1, 0]. As an illustration of this binary fix, Figure 3 shows feasible and infeasible perturbations from a point • with ○ and × respectively. The binaries of the non-convex minimum speed constraints are fixed in the same way to allow for maximum perturbation, as shown in Figure 4.

Note that the problem statement includes many binary variables that are effectively fixed or constraints that are always satisfied (e.g. the lower left side of constraints in Figure 4). The MILP solver CPLEX eliminates these redundant variables and constraints in the pre-solve step, and the size of the CDRSBK subproblem increases only slightly from the DRSBK subproblem. In the example shown later in Section VII-C, the number of variables after the CPLEX pre-solve increased by 15% for continuous variables and 1% for binary variables.

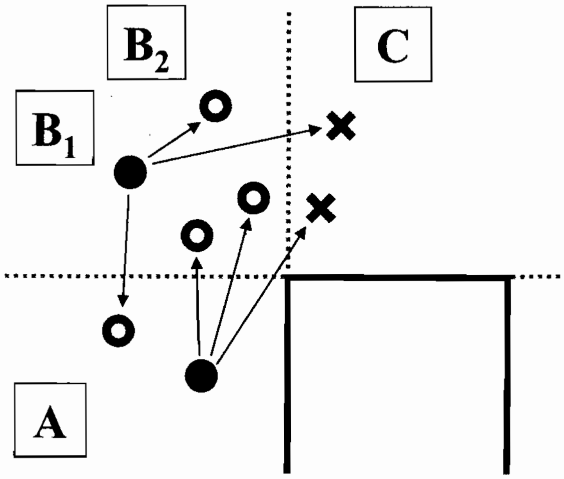


Fig. 3. Feasible perturbation of the position, with fixed obstacle avoidance binaries

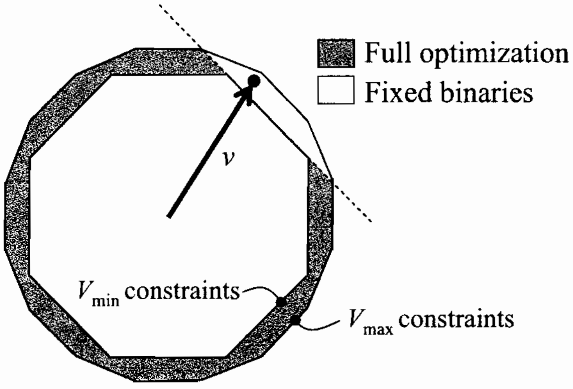


Fig. 4. Feasible perturbation of the velocity, with fixed $v_{\min}$ binaries

For the fixed-wing aircraft, when the terminal safety constraint requires the plan to end in a loiter circle, there is one exception in the binary fix. The location of the safety circle has a large effect on the objective value, and therefore it is beneficial to keep the binary variable that selects left or right circle (b^q_{left} in (44) in Appendix) as a decision variable in p's optimization.

C. Subproblem

This subsection gives a brief explanation of the subproblem that each vehicle solves. The Appendix gives a detailed implementation using MILP. Let $\tilde{\mathsf{C}}^p(\boldsymbol{x}^p_k, U^p_k)$ denote a set of constraints that is the same as $\mathsf{C}^p(\boldsymbol{x}^p_k, U^p_k)$ except that the tightened constraint equations (9) and (11) are replaced by the following

$$\forall j: \qquad \boldsymbol{y}^p_{k+j|k} \in \mathcal{Y}^p_{j+1} \tag{28}$$

$$\boldsymbol{z}^p_{k+j|k} + \tilde{\boldsymbol{z}}^p_{k+j|k} \in \mathcal{Z}^p_{j+1} \tag{29}$$

with $\mathcal{Y}^p_N \triangleq \mathcal{Y}^p_{N-1}$ and $\mathcal{Z}^p_N \triangleq \mathcal{Z}^p_{N-1}$, which are consistent with the tightening equations (14) and (16) under the nilpotency assumption $L^p_{N-1} = 0$. Then, the vehicle p solves the following optimization P^p_k:

$$\begin{aligned} &\min_{\xi^1,\dots,\xi^{p-1},U^p_k,\xi^{p+1},\dots,\xi^n} \quad J \\ &\text{s.t. } (24), (25) \text{ or } (27), \\ &\quad F\Big(\mathsf{C}^1(\boldsymbol{x}^1_k, \bar{U}^1_k + \delta U^1_k), \dots, \mathsf{C}^{p-1}(\boldsymbol{x}^{p-1}_k, \bar{U}^{p-1}_k + \delta U^{p-1}_k),\ \mathsf{C}^p(\boldsymbol{x}^p_k, U^p_k), \\ &\qquad \tilde{\mathsf{C}}^{p+1}(\boldsymbol{x}^{p+1}_{k|k-1}, \bar{U}^{p+1}_k + \delta U^{p+1}_k), \dots, \tilde{\mathsf{C}}^n(\boldsymbol{x}^n_{k|k-1}, \bar{U}^n_k + \delta U^n_k)\Big) \leq 0 \end{aligned} \tag{30}$$

The function $F(\cdot)$ represents the constraints imposed on all vehicles. In this optimization, vehicle p designs its own control U^p_k as well as perturbation to other vehicles' control $\delta U^1_k, \dots, \delta U^{p-1}_k$, $\delta U^{p+1}_k, \dots, \delta U^n_k$. The notation $(\bar{\cdot})^q$ denotes the variables that are received from other vehicles and are constant in vehicle p's subproblem, so that the control inputs for vehicle q are

$$U^q_k = \bar{U}^q_k + \delta U^q_k. \tag{31}$$

As shown in (30), if a vehicle q has already planned at time k, i.e., $q < p$, then the vehicle p uses $\mathsf{C}^q(\boldsymbol{x}^q_k, \bar{U}^q_k + \delta U^q_k)$ as the constraints for q. Otherwise, the constraints for q are $\tilde{\mathsf{C}}^q(\boldsymbol{x}^q_{k|k-1}, \bar{U}^q_k + \delta U^q_k)$, where the initial states are the states predicted in the previous plan. The objective is to minimize the fleet cost (4). After solving the optimization,

Algorithm 1 Cooperative DRSBK

```
1: Find a feasible solution of the DRSBK optimization starting from the initial states $\boldsymbol{x}_0^p$, communicate the
   solutions, form $\bar{U}_0^1, \dots, \bar{U}_0^n$, and set $k = 1$.
2: for $k = 1$ to $k = \infty$ do
3:   Form the candidate control $\bar{U}_k^1, \dots, \bar{U}_k^n$ from the previous solution $\bar{U}_{k-1}^1, \dots, \bar{U}_{k-1}^n$ using (32).
4:   repeat
5:     for $p = 1, \dots, n$ do
6:       Measure the current states and form $\boldsymbol{x}_k^p$.
7:       Solve the subproblem $\mathsf{P}_k^p$ and obtain the solution $U_k^{p*}$, $\xi^{q*}$.
8:       Update the control $\bar{U}_k^p := U_k^{p*}$, $\bar{U}_k^q := \bar{U}_k^q + T^q \xi^{q*}$.
9:       Send the solutions to the next vehicle.
10:    end for
11:  until converges or reaches the iteration limit
12:  Apply the first step of the control inputs $\boldsymbol{u}_k^p = \bar{\boldsymbol{u}}_{k|k}^p$ to the system (1), $\forall p$.
13: end for
```

the solution $(\cdot)^*$ updates the control as

$$\begin{aligned} \bar{U}_k^p &:= U_k^{p*} \\ \bar{U}_k^q &:= \bar{U}_k^q + \delta U_k^{q*}, \quad \forall q \neq p \end{aligned}$$

which are sent to the next vehicle $p+1$.

The full CDRSBK algorithm is shown in Algorithm 1. In the line 7 of Algorithm 1, the vehicle p makes a decision on itself and its neighbors. As shown in line 3, when the time step is incremented, the candidate decisions $\bar{U}_k^p$ for the current time are constructed from the decisions $\bar{U}_{k-1}^p$ made at the previous time, using

$$\forall p: \qquad \bar{\boldsymbol{u}}_{k+j|k}^p = \bar{\boldsymbol{u}}_{k+j|k-1}^p, \qquad \forall j^- \tag{32a}$$

$$\bar{\boldsymbol{u}}_{k+N-1|k}^p = \kappa^p(\bar{\boldsymbol{x}}_{k+N-1|k-1}^p). \tag{32b}$$

This operation shifts the plan $\bar{U}_{k-1}^p$ by one time step and appends the terminal step taken from the invariance control law $\kappa^p(\boldsymbol{x})$ in (17). The states $\bar{\boldsymbol{x}}_{k+N-1|k-1}^p$ are obtained through the state equation

$$\bar{\boldsymbol{x}}_{k+N+1|k-1}^p = A^p \bar{\boldsymbol{x}}_{k+N|k-1}^p + B^p \bar{\boldsymbol{u}}_{k+N|k-1}^p.$$

Note that the local optimization (19) requires the knowledge of other vehicles' cost function, but this depends only on the target score, target location, vehicle states, etc., and is simple to communicate.

D. Implementation with Non-zero Computation Time

To run the algorithm in real-time on the vehicle hardware, non-zero computation and communication times must be handled. This is done by propagating the measured states when forming $\boldsymbol{x}_k^p$ in line 6 of Algorithm 1. Figure 5 shows the time lines of the CDRSBK algorithm implementation.

The discrete time step k is defined as the time when the control $\boldsymbol{u}_{k|k}$ is executed in line 12 of Algorithm 1. The latest measurement is taken τ seconds before the discrete time, where τ is an upper bound on the computation and communication time to the discrete time. As shown with the gray arrow, the vehicle propagates the measured states up to the discrete execution time using a model of the vehicle and the low-level controller. The propagated states $\boldsymbol{x}_{k|k}$ are then used as the initial condition of the optimization at time k. When the last vehicle finishes its computation, it broadcasts the final plans to the fleet, and all vehicles implement the control at the same time. Note that when the vehicle p is planning at time k, the prediction for vehicle $p-1$ is based on the latest states $\boldsymbol{x}_{k|k}^{p-1}$ and the inputs $\boldsymbol{u}_{\cdot|k}^{p-1}$, but the prediction for vehicle $p+1$ is based on the states $\boldsymbol{x}_{k|k-1}^{p+1}$ and the inputs $\boldsymbol{u}_{\cdot|k-1}^{p-1}$ that are predicted at the previous time $k-1$.

By aligning the states and the control inputs of all vehicles at the discrete time steps, this framework compiles various sources of uncertainties into one and allows us to treat them as a single prediction error, which can be

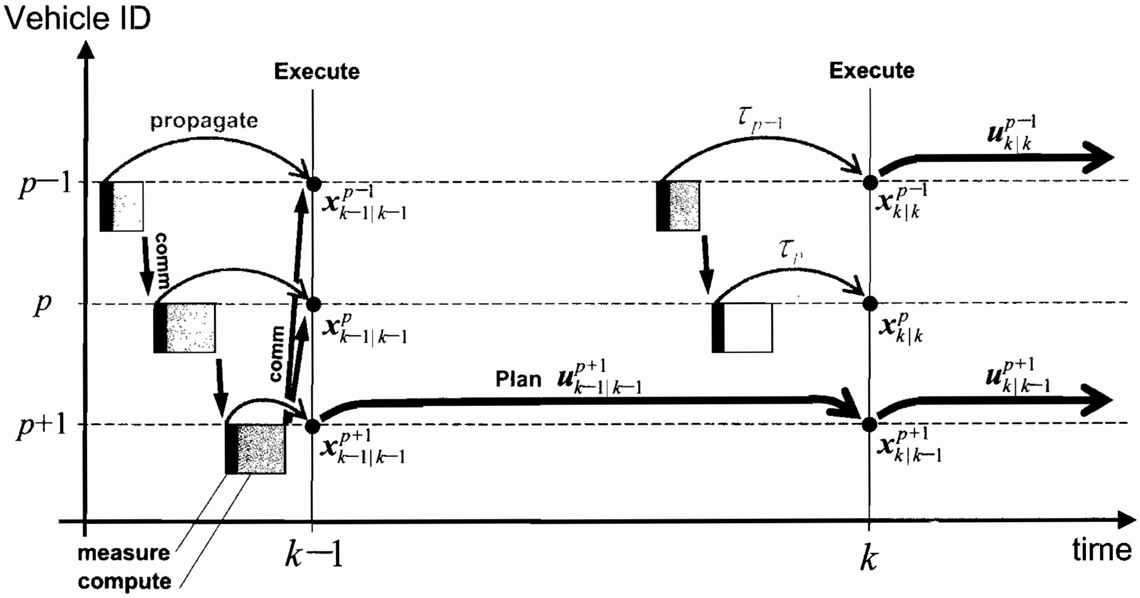


Fig. 5. Time flow of CDRSBK algorithm with non-zero computation and communication time

calculated simply as $w^p_{k-1} = x^p_{k|k} - x^p_{k|k-1}$. Future work will extend this implementation to reduce the delay associated with the propagation, as studied in [31].

VI. Algorithm Properties

This section discusses the two important properties of CDRSBK algorithm in terms of constraint satisfaction and the performance.

A. Robust Feasibility

The following theorem guarantees the robust feasibility of the entire fleet after solving each subproblem.

Theorem 1: If a feasible solution $\bar{U}^1_0, \dots, \bar{U}^n_0$ to the following constraint ((30) with $k = 0,\ p = n$) is found

$$F\Big(\mathrm{C}^1(x^1_0, \bar{U}^1_0), \dots, \mathrm{C}^n(x^n_0, \bar{U}^n_0)\Big) \le 0 \tag{33}$$

then, the system (1) controlled by Algorithm 1 satisfies the constraints (2)–(3) under the action of disturbance $w^p_k \in \mathcal{W}^p$ for all vehicles p and all future times $k\,(> 0)$.

Proof: The proof builds on the robust feasibility property of DRSBK algorithm, using the fact that the CDRSBK algorithm reduces to the DRSBK algorithm when no perturbations are allowed. In DRSBK, it was shown in [20] that given a feasible solution at time $k-1$, the following candidate solution

$$\hat{u}^p_{k+j|k} = \bar{u}^p_{k+j|k-1} + P^p_{j+1} w^p_{k-1}, \quad \forall j^- \tag{34a}$$

$$\hat{x}^p_{k+j|k} = \bar{x}^p_{k+j|k-1} + L^p_j w^p_{k-1}, \quad \forall j \tag{34b}$$

$$\hat{u}^p_{k+N-1|k} = \kappa^p(\hat{x}^p_{k+N-1|k}), \tag{34c}$$

$$\hat{x}^p_{k+N|k} = A^p \hat{x}^p_{k+N-1|k} + B^p \hat{u}^p_{k+N-1|k} \tag{34d}$$

is feasible to vehicle p's optimization at time k for any disturbance realization w^p_{k-1}. Therefore, in CDRSBK, using (34) for its own decision and using

$$\delta U^q_k = 0, \quad \forall q \tag{35}$$

for other vehicles' decisions form a feasible candidate solution for p's optimization. This shows the feasibility at time $k-1$ implies the feasibility at time k. By recursion, if the optimization at time $k=0$ is feasible, all future optimization is feasible, which means the vehicles satisfy all the constraints. ■

Remark 1 (Anytime algorithm): Because the entire vehicle fleet retains feasibility after solving each subproblem, the algorithm can be stopped at any time in the iteration.

B. Monotonic Decrease of the Fleet Cost

Another important property of CDRSBK algorithm is that the fleet objective is monotonically decreasing by solving each subproblem.

Theorem 2: The fleet objective value monotonically decreases by solving each subproblem P_k^p over the fleet (line 5 in Algorithm 1) and over the time (line 2).

Proof: The proof is based on showing that the candidate solution to p's optimization yields an objective value that is no worse than the objective value found in the optimization of the previous vehicle $p-1$ (or n if $p=1$).

In p's subproblem, the candidate solution (35) does not change variables for vehicles $\forall q$, and hence yields the same local cost for other vehicles J^q that was found in the optimization by $p-1$. For the cost of vehicle p itself, with the assumption of nilpotency $L_{N-1}^p = 0$, the terminal states can be written as

$$\hat{\boldsymbol{x}}_{k+N|k}^p = A^p \bar{\boldsymbol{x}}_{k+N-1|k-1}^p + B^p \kappa^p (\bar{\boldsymbol{x}}_{k+N-1|k-1}^p)$$

using (34b)–(34d). The individual objective value J^p depends only on the point $\boldsymbol{O}^p$ in the invariant, which can remain at the same location using the invariant terminal controller $\kappa^p(\cdot)$ (e.g. loitering in a circle with a constant turn radius, or hovering). This implies the candidate solution (34) produces the same individual cost J^p obtained at the previous optimization. Because p's optimization can use the candidate solution to achieve the same fleet cost obtained by the previous vehicle $(p-1)$, the fleet cost can only improve by optimization. ■

Note that this does not mean the individual cost decreases monotonically. The simulation results in Section VII and VIII demonstrate a temporary increase of the individual cost that leads to a greater reduction of the fleet cost.

VII. Simulation Results

A. Vehicle Model

The simulation uses homogeneous n vehicles. A point-mass model is used to approximate the translational dynamics of UAVs

$$\begin{bmatrix} \boldsymbol{r}_{k+1}^p \\ \boldsymbol{v}_{k+1}^p \end{bmatrix} = A^p \begin{bmatrix} \boldsymbol{r}_k^p \\ \boldsymbol{v}_k^p \end{bmatrix} + B^p \boldsymbol{a}_k^p + \boldsymbol{w}_k^p \tag{36a}$$

$$A^p = \begin{bmatrix} I_2 & \Delta t\, I_2 \\ O_2 & I_2 \end{bmatrix}, \quad B^p = \begin{bmatrix} \frac{(\Delta t)^2}{2} I_2 \\ \Delta t I_2 \end{bmatrix} \tag{36b}$$

where $\boldsymbol{r}^p$, $\boldsymbol{v}^p$, and $\boldsymbol{a}^p$ are the position, the velocity, and the acceleration vector respectively. Matrices I_2 and O_2 express an identity matrix and a zero matrix of size 2 respectively. The disturbance $\boldsymbol{w}_k^p$ enters through the input

$$\boldsymbol{w}_k^p \in \mathcal{W}^p = \left\{ \boldsymbol{w} \;\middle|\; \boldsymbol{w} = B^p \boldsymbol{n},\ \boldsymbol{n} \in \mathbb{R}^2,\ \|\boldsymbol{n}\|_\infty \le w_{\max} \right\}. \tag{37}$$

The local constraints include the obstacle avoidance, the maximum/minimum speed, and the maximum input constraints

$$\begin{aligned} \boldsymbol{r}_k^p &\notin \mathcal{O}, \\ v_{\min} \le \|\boldsymbol{v}_k^p\|_2 &\le v_{\max} \\ \|\boldsymbol{a}_k^p\|_2 &\le a_{\max} \end{aligned}$$

where $\mathcal{O} \subset \mathbb{R}^2$ expresses the no-fly zones, and $v_{\min}, v_{\max}, a_{\max}$ are the minimum speed, maximum speed, and maximum acceleration of the vehicle.

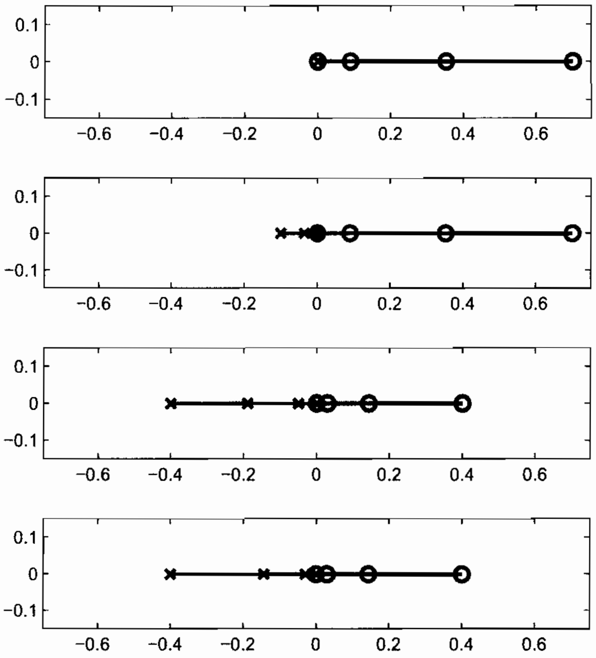


Fig. 6. The evolution of plans over the iteration for the simple two vehicle example.

B. Iteration Within Each Time Step

The first set of simulation is based on the simplified problem setup to evaluate the solution within a single time step. Only the convex constraints are considered, so no binary variables are used. Furthermore, zero disturbance is assumed and hence no constraints are tightened.

1) Simulation Setup: In this simulation, Δt in (36) is set to 1. The parameters for the constraints are $v_{\text{min}} = 0, v_{\text{max}} = 0.35, a_{\text{max}} = 0.18$, and $\|\boldsymbol{r}_k^p\|_2 \leq 1$. The systems are coupled with two neighbors through the following position constraints.

$$\left\|\boldsymbol{r}_k^p - \boldsymbol{r}_k^{p+1}\right\|_2 \leq 0.8, \qquad p = 1, \dots, n-1$$
$$\left\|\boldsymbol{r}_k^n - \boldsymbol{r}_k^1\right\|_2 \leq 0.8$$

These two-norm constraints are expressed as a combination of linear constraints. The cost direction for the p^{th} vehicle is

$$c_p = \begin{bmatrix} \cos\left(\frac{p-1}{n}\right) & \sin\left(\frac{p-1}{n}\right) \end{bmatrix}^T.$$

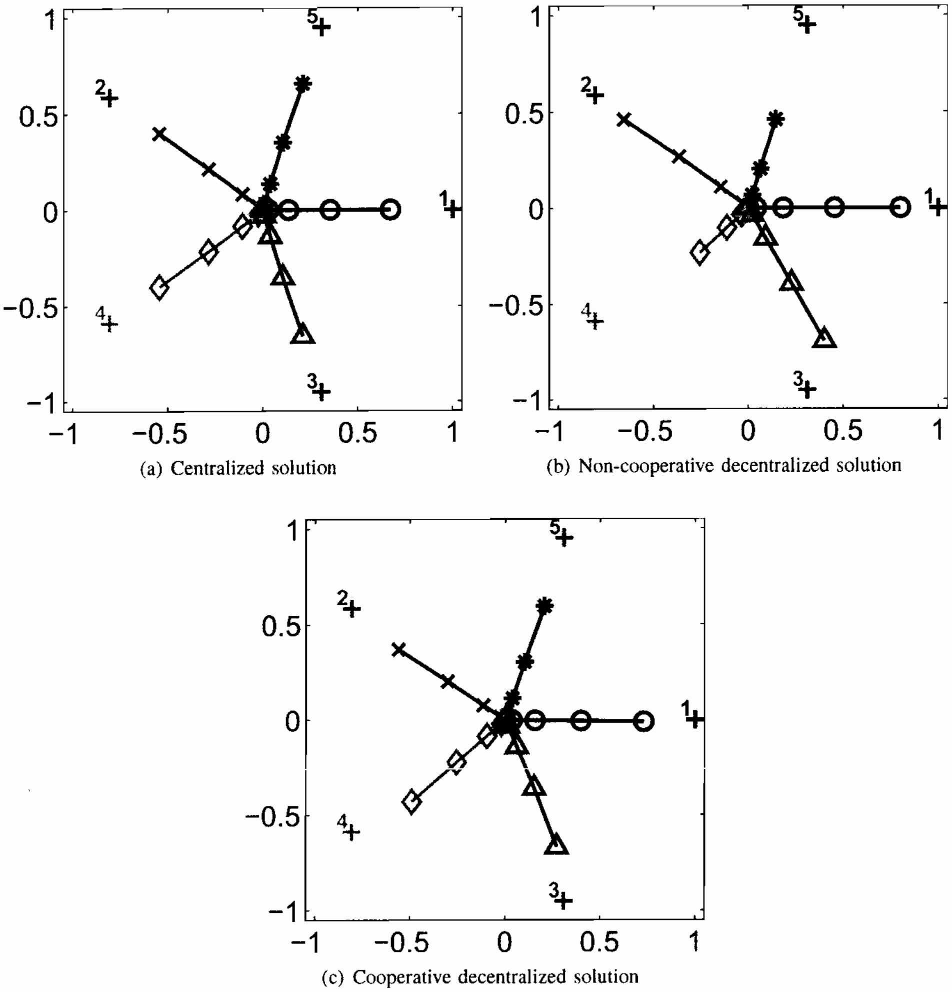
The overall cost function to minimize is quadratic, and has a stage cost and the terminal penalty

$$\sum_{p=1}^{n}\sum_{k=0}^{N-1}\left\{\boldsymbol{x}_k^{p\,T} R_1 \boldsymbol{x}_k^p + \boldsymbol{u}_k^{p\,T} R_2 \boldsymbol{u}_k^p\right\} + c_p{}^T \boldsymbol{r}_N^p + \boldsymbol{r}_N^p{}^T H \boldsymbol{r}_N^p$$

where the weights on the states R_1 and inputs R_2 in the stage cost are chosen to be much smaller than the weight H on the terminal position. All the vehicles are initially at the origin. Both the centralized and the local optimization are written as quadratic programming, and CPLEX 9.1 is used as a solver.

2) Simple Two Vehicle Case: The first example involves two vehicles p and q that can move on a two dimensional plane. The terminal position of the vehicle p has its local minimum at coordinates (0, 0.7), i.e.,

$$\begin{bmatrix} 0 \\ 0.7 \end{bmatrix} = \arg\min_{\boldsymbol{r}_N^p}\left\{c_i{}^T \boldsymbol{r}_N^p + \boldsymbol{r}_N^p{}^T H \boldsymbol{r}_N^p\right\},$$



(a) Centralized solution

(b) Non-cooperative decentralized solution

(c) Cooperative decentralized solution

Fig. 7. Final plans for five vehicles

and that of the vehicle q is at $(0, -0.7)$. Because the two vehicles must satisfy the separation constraint of 0.8, their separate objectives are conflicting. The planning horizon is three steps for both vehicles.

Figure 6 shows the evolution of plans over two iterations. The plans of vehicle p are marked with $\circ$, and those of vehicle q are marked with $\times$. Originally, both vehicles are at the origin. First, vehicle p solves its local optimization. Because no coupling constraints are active at this point, the terminal position reaches the local minimum (0, 0.7). Vehicle q then solves its optimization, but given the separation constraint, this vehicle can only plan to move to $(0, -0.1)$, as shown in the second part of the figure.

The vehicle p solves the next optimization, but since a coupling constraint has become active, it uses a parameterized decision for q with a variable ξ^q of dimension $m = 1$. The bottom figure shows the plans after two iterations. The final plans are the same as the globally optimal centralized solution.

If the decentralized non-cooperative algorithm in Section IV were used, it would produce a Pareto optimal solution shown in the second figure of Figure 6, which is clearly not the globally optimal solution shown in the bottom. Note that if the vehicle q plans first followed by vehicle p, the non-cooperative algorithm results in a symmetric Pareto optimal solution, which again is not the globally optimal solution. This example clearly shows the performance improvement over the decentralized non-cooperative approach.

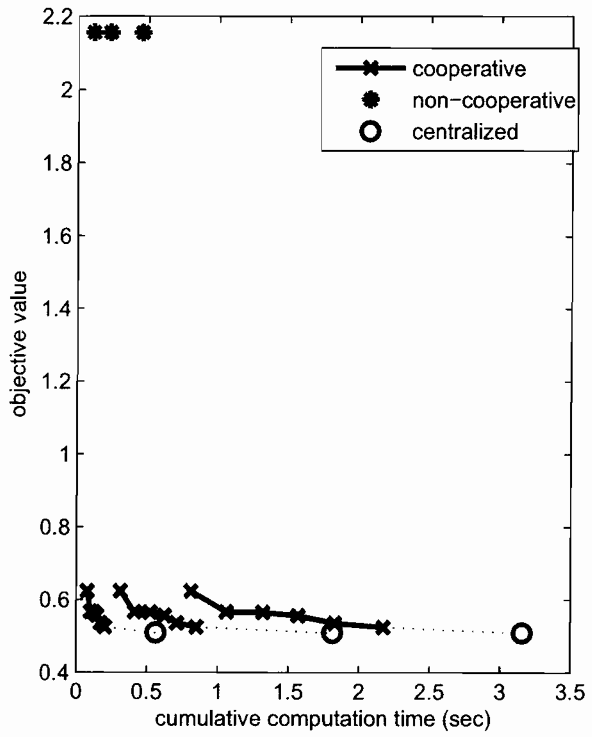


Fig. 8. Comparison of three algorithms in terms of performance and computation.

3) Five Vehicle Case: Figure 7 shows a more complex case with five vehicles. In this example, the local minimum for each vehicle is located on a unit circle centered at the origin. The planning horizon is three steps for all vehicles, and the planning order is $1 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 4$ to highlight the effect of the planning order on the performance.

Two other algorithms are used as benchmarks. These are: 1) the centralized approach in Section III that provides the globally optimal solution, and 2) the decentralized non-cooperative approach in Section IV that produces a locally optimized solution.

As shown in Figure 7(b), the decentralized non-cooperative approach produced a suboptimal solution, because the vehicles that plan earlier are less constrained and have more region to operate than the vehicles that plan later in the cycle. The decentralized cooperation algorithm produced the trajectories shown in Figure 7(c) whose shape are very similar to the centralized solution shown in Figure 7(a).

4) Performance and Computation: Figure 8 compares the global objective value and the cumulative computation time of three algorithms for the five vehicle example. Different lengths of the planning horizon $N = 4, 6, 8$ were considered to investigate the scalability of the algorithms.

The solutions of the decentralized non-cooperative approach are marked with $*$. Although the computation time is small, the cost is fairly high. The centralized (and hence globally optimal) solutions are marked with $\circ$. The lines with $\times$ show the evolution of the global cost of the decentralized cooperation algorithm. The plot starts from the end of the first iteration when every vehicle has its solution and continues to the end of the second iteration. This proposed algorithm has objective values comparable to those of the centralized solution but scales better than the centralized solution when the problem size increases.

Figure 9 shows cases with more vehicles (n = 5, 7, 10, 15). The decentralized non-cooperative approach has much higher cost and is out of the range of the plot. For the centralized and the proposed approach, the differences in the computation time scale up significantly for larger fleets. Note that in all the plots of Figures 8 and 9, the lines of the proposed approach are monotonically decreasing, which validates the result in Section VI-B by simulation.

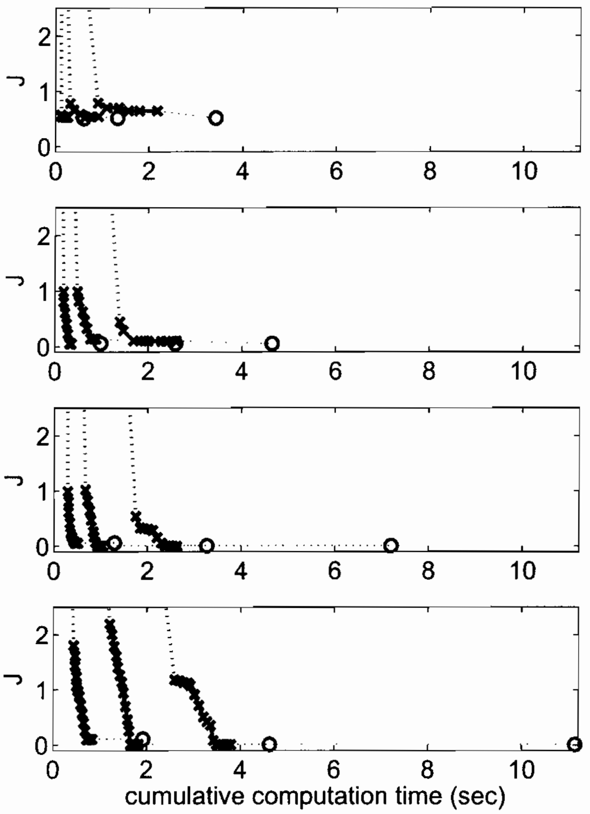


Fig. 9. Trade-off between the performance and the computation time. From the top to the bottom, the number of vehicles are 5, 7, 10, and 15.

C. Full CDRSBK Algorithm

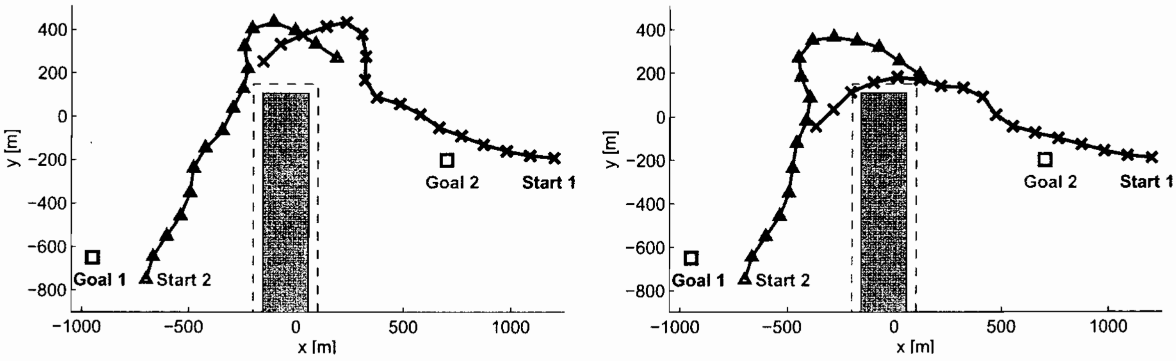
This section presents the performance comparison of DRSBK and CDRSBK through simulation.

1) Setup: The simulation uses fixed-wing UAVs, whose dynamics are (36) with $\Delta t = 5$. The disturbance magnitude w_{max} is 5% of the control authority a_{max}. The planning horizon length N is 4. The parameters for dynamic constraints are: $v_{\text{min}} = 18$, $v_{\text{max}} = 24$, $a_{\text{max}} = 3.84$. A two-step nilpotent controller is used for this system to tighten the constraints in (14) and (16), and the parameters for constraint tightening are obtained through (39) in the Appendix

$$\begin{array}{llll} \alpha_0 = 0, & \beta_0 = 0, & \gamma_0 = 0, & \\ \alpha_1 = 2.4, & \beta_1 = 1.4, & \gamma_1 = 0.54, & \\ \alpha_j = 4.8, & \beta_j = 2.7, & \gamma_j = 0.81, & j \geq 2. \end{array}$$

When the coupling constraints are active, the parameterization matrix T^q is calculated from (25)

$$T^q = \begin{bmatrix} 0.012 & 0 & -0.07 & 0 \\ 0 & 0.012 & 0 & -0.07 \\ 0.004 & 0 & 0.01 & 0 \\ 0 & 0.004 & 0 & 0.01 \\ -0.004 & 0 & 0.09 & 0 \\ 0 & -0.004 & 0 & 0.09 \\ -0.012 & 0 & 0.17 & 0 \\ 0 & -0.012 & 0 & 0.17 \end{bmatrix}$$



(a) DRSBK algorithm

(b) CDRSBK algorithm

Fig. 10. Comparison of trajectories executed.

showing the "tall" parameterization matrix that reduces the dimension of q's decision space. Although the objective function can be an arbitrary function of all vehicles' decisions, the goal of the trajectory optimization is assumed to minimize the mission completion time of the fleet in this example. The summation of the individual cost is also included with a small penalty ϵ, so that the vehicles that complete the mission before the last vehicle also minimize the individual completion time.

$$J\left(J^0,\dots,J^n\right) = \max_p J^p(U^p) + \epsilon \sum_{p=1}^{n} J^p(U^p) \tag{38}$$

2) Results: The scenario considers two vehicles trying to reach their own targets (marked with □) while avoiding obstacles and the other vehicle. The goal is to minimize the mission completion time with a small penalty $\epsilon = 0.05$ on the individual cost in (38). Figure 10 shows the trajectories generated by DRSBK and CDRSBK algorithms. The trajectory of vehicle 1 is marked with $\times$, and that of vehicle 2 is marked with $\triangle$. Because vehicle 1 has to traverse a longer route, the optimal solution is for vehicle 2 to move over, which is the behavior achieved by the CDRSBK algorithm. Both distributed algorithms maintained feasibility under the action of disturbances. However, DRSBK subproblem solely minimizes the individual cost without considering the performance of the other vehicle, making no improvement on the cost for some time.

This cooperative behavior is also seen in the plot for the objective values. Figure 11 shows the time history of the individual cost J^p and the fleet cost J. Both algorithms monotonically decrease the fleet objective. As shown in the right figure, the cooperative formulation allows the individual cost to increase if it leads to a larger improvement of the fleet cost. Between optimization #14–17 (which correspond to time 7–9), the vehicle with a better cost (vehicle 2) yields to the vehicle with a worse cost (vehicle 1), enabling a large reduction in the fleet cost J. The average computation time for solving MILP in this scenario was 0.050 second for DRSBK and 0.064 second for CDRSBK.

VIII. Hardware Results

A similar scenario is tested using an unique quadrotor testbed developed at the Aerospace Controls Laboratory of MIT.

A. Quadrotor Testbed

This section briefly describes the quadrotor testbed. More details are available in the recent article [32]. As a sensing device, the testbed uses a Vicon motion capture system [34], which are the cameras shown in background of Figure 12. The Vicon system provides a position estimate of sub-millimeter accuracy at 100 Hz, and the filtered time difference gives a velocity estimate of 1 cm/s peak-to-peak accuracy. The vehicles are commercially available Draganflyer V Ti Pro [35], and no significant modifications was required to the hardware to fly them autonomously. Several lightweight reflective markers are attached to each vehicle in a unique configuration, which the Vicon system uses to track the position and orientation of each vehicle in the room. The low-level controller is designed to track

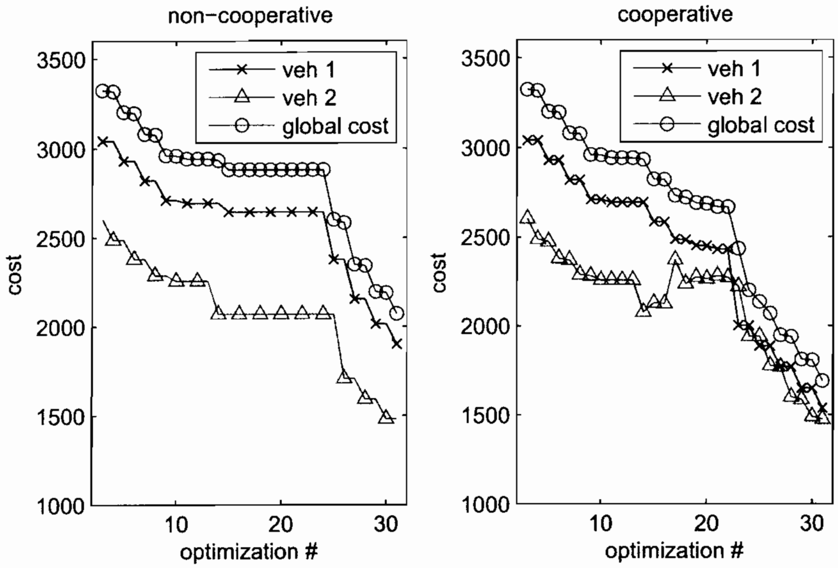


Fig. 11. Time history of the objective function

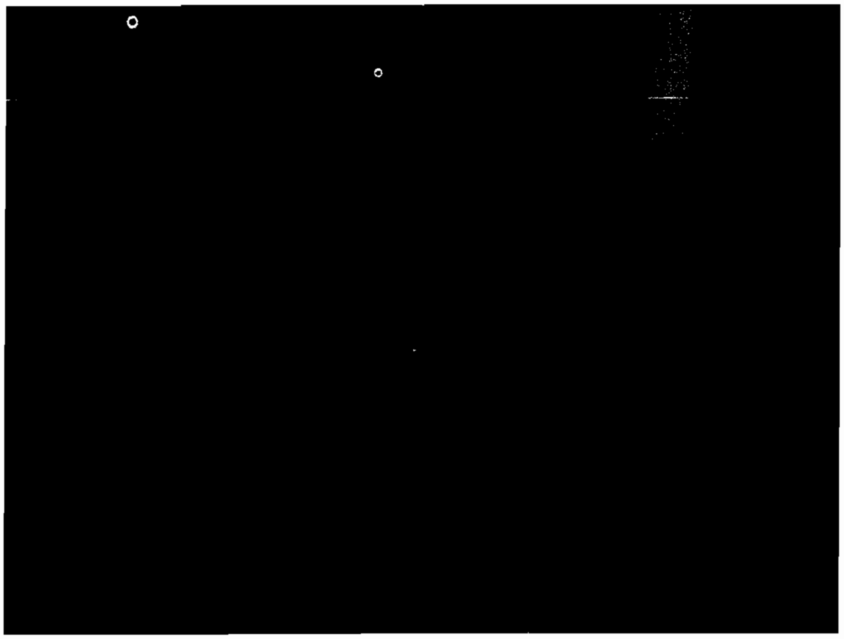
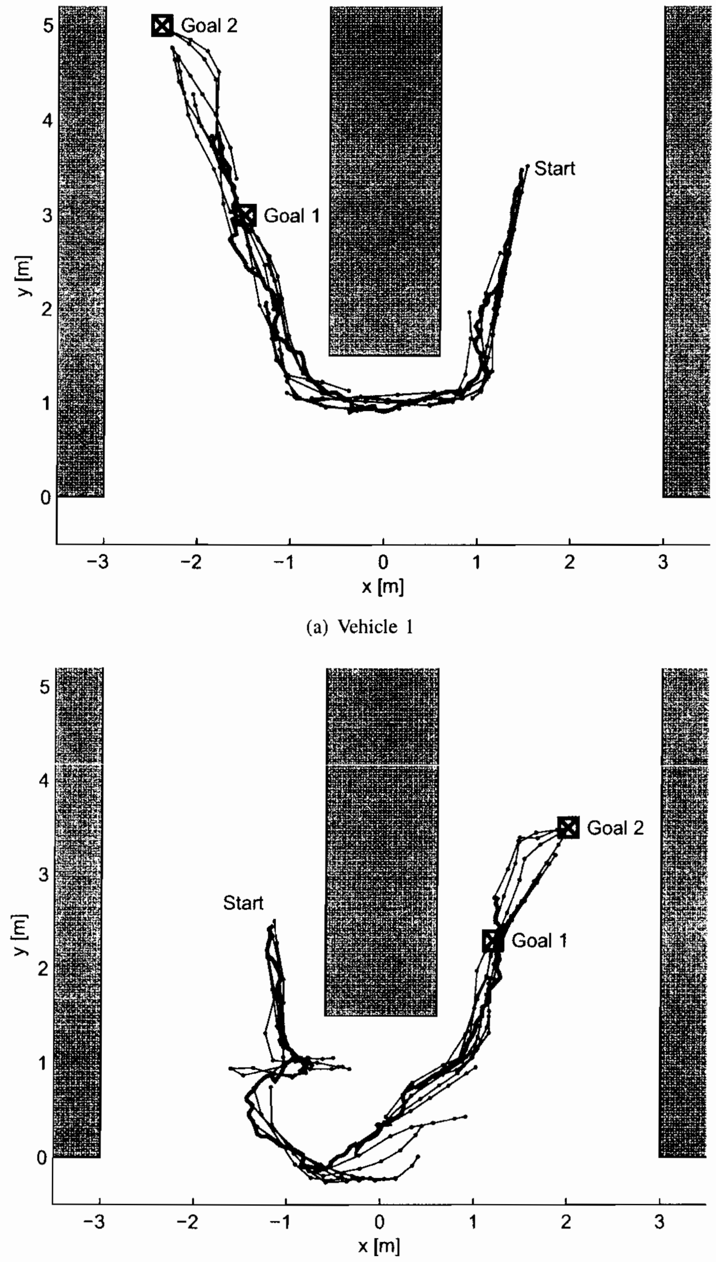


Fig. 12. Quadrotor testbed using Vicon system [32], [33]

waypoints which are provided in real-time by the planner. The waypoint follower was designed using standard LQR techniques, which calculates the motor commands of each rotor off-board and sends them to the quadrotor using an R/C transmitter [32].

A planning laptop is assigned to each vehicle, and the inter-vehicle communication is implemented as a communication over the TCP/IP network.



(a) Vehicle 1

(b) Vehicle 2

Fig. 13. The plans and the trajectories of two quadrotors from the CDRSBK algorithm experiment

B. Result

The objective of this experiment is to demonstrate that the algorithm can generate online a trajectory using receding horizon techniques. Many disturbances sources exist in the hardware experiments, such as air flow, modeling error of the vehicle, sensing noise, communication delay, and imperfect tracking of the low-level controller. The CDRSBK algorithm must account for these uncertainties to robustly satisfy the constraints. The following parameters were

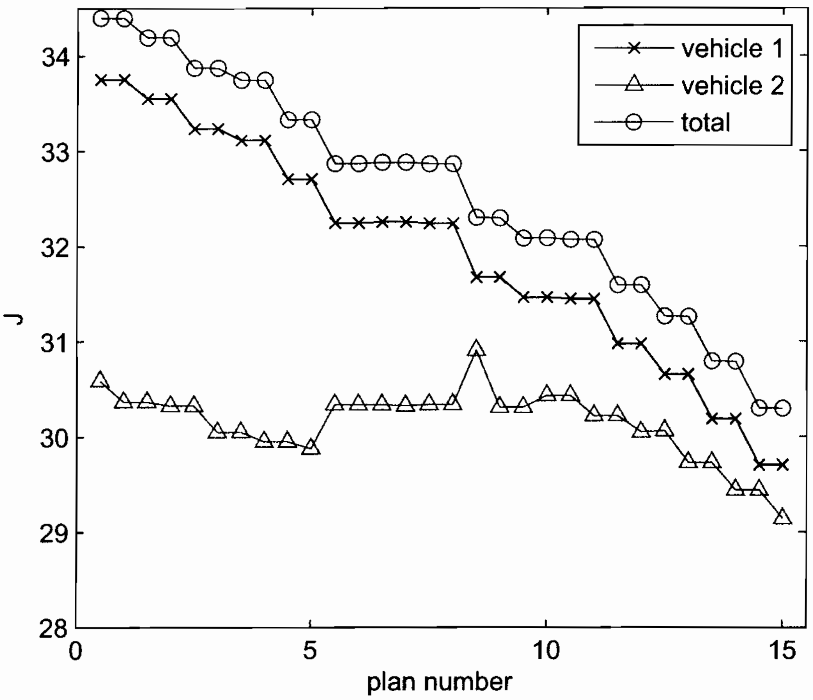


Fig. 14. Individual costs and the global cost

used in the algorithm.

$$\Delta t = 2.5\,\text{sec}, \qquad N = 6$$

$$\tau_1 = 1.3\,\text{sec}, \qquad \tau_2 = 0.7\,\text{sec}$$

$$v_{\text{max}} = 0.30\,\text{m/s}, \qquad a_{\text{max}} = 0.45\,\text{m/s}^2$$

$$w_r = 0.27\,\text{m}, \qquad w_v = 0.09\,\text{m/s}$$

where τ_1 and τ_2 are the propagation time for vehicle 1 and 2, as shown in Figure 5. Here, the disturbance enters separately for position and velocity. Because the quadrotors can hover, the full stop was used as the terminal safety constraints. The procedure in [26] produced the following constraint contraction parameters in (39).

$$\alpha_0 = 0, \qquad \beta_0 = 0, \qquad \gamma_0 = 0,$$

$$\alpha_1 = 0.27, \qquad \beta_1 = 0.09, \qquad \gamma_1 = 0.067,$$

$$\alpha_2 = 0.428, \qquad \beta_2 = 0.198, \qquad \gamma_2 = 0.106,$$

$$\alpha_j = 0.440, \qquad \beta_j = 0.208, \qquad \gamma_j = 0.110, \qquad j \geq 3$$

The vehicles 1 and 2 started around $(1.5, 3.5)$ and $(-1.2, 2.5)$ respectively. The targets for vehicle 1 are $(-1.5, 3.0)$ are $(-2.4, 5.0)$, and the targets for vehicle 2 are $(1.2, 2.3)$ and $(2.0, 3.5)$, which are all marked with ☒. The vehicles must switch the position while avoiding the other vehicle and the obstacle in the middle.

Figure 13 shows this scenario and the plot of the trajectory of each vehicle. The red thick line is the actual trajectory of the vehicle, which were recorded at 2 Hz. All the plans generated in the receding horizon framework are shown with blue lines.

As a result of their initial locations, vehicle 2 approaches the bottom of the obstacle before vehicle 1, and vehicle 2 then tries to go along the bottom of the obstacle, as the planned trajectories in Figure 13(b) show. This would have the effect of delaying vehicle 1, which has targets that are further away, and already has a longer mission to execute than vehicle 2. Thus, vehicle 2 yields way to vehicle 1 to minimize the fleet mission completion time in (38). This cooperative effect is also shown in Figure 14, which plots the objective values of the first 15 plans. Note that between plans 5 and 9, there is a temporary increase of the cost for vehicle 2, but, even under the action of disturbances, the total objective value is monotonically decreasing. The average computation time for each local optimization on a 2.4 GHz laptop was 0.31 second. This flight test successfully demonstrated the cooperative behavior by the distributed online planning algorithm.

IX. Conclusions

This paper presented a robust decentralized trajectory optimization algorithm that includes explicit cooperation. Each vehicle sequentially solves the subproblem, but the subproblem also includes the global objective and feasible modifications to other vehicles' plans. The overall optimization is written in MILP. In order to maintain the scalability of the algorithm, continuous variables of neighboring vehicles are parameterized using a variable of smaller dimension, while most binary variables are fixed. It is shown to guarantee the robust feasibility and the monotonic decrease of the global cost. Simulation results showed that the proposed algorithm scales much better than the centralized approach and the performance is much better than that of the non-cooperative approach with a marginal increase in the computation. The algorithm was implemented on the quadrotor testbed, and the various features of the algorithm have been successfully demonstrated.

Appendix

The following describes the MILP implementation of the CDRSBK algorithm. In this section, the disturbance is assumed to be infinity-norm bounded, i.e., $\mathcal{W}^p = \left\{Gw \mid \|w\|_\infty \le w_{\max}\right\}$

A. Constraint Tightening for Robustness

The constraint tightening in (14) and (16) are implemented by using the following constraint contraction parameters [36]

$$\begin{aligned}
\alpha_0 &= 0, & \alpha_j &= \alpha_{j-1} + \left\|[1\ 0\ 0\ 0]L^p_{j-1}G\right\|_1 w_{\max},\ j \ge 1 \\
\beta_0 &= 0, & \beta_j &= \beta_{j-1} + \left\|[0\ 0\ 1\ 0]L^p_{j-1}G\right\|_1 w_{\max},\ j \ge 1 \\
\gamma_0 &= 0, & \gamma_j &= \gamma_{j-1} + \left\|[1\ 0]K^p_{j-1}L^p_{j-1}G\right\|_1 w_{\max},\ j \ge 1
\end{aligned} \tag{39}$$

where α_j, β_j, and γ_j respectively represents the constraint contraction for position, velocity, and input for the j^{th} prediction step.

B. Output Constraint Set (9)

For each point $[x, y]^T$ and each rectangular shaped obstacle defined by two corners $[x_{\text{low}}, y_{\text{low}}]^T$ and $[x_{\text{high}}, y_{\text{high}}]^T$, the obstacle avoidance constraints can be written using binary variables

$$\forall o,\ \forall j: \quad x^p_{k+j|k} \le x_{\text{low},o} - \alpha_j + M\, b^p_{\text{obst},jo1} \tag{40a}$$

$$y^p_{k+j|k} \le y_{\text{low},o} - \alpha_j + M\, b^p_{\text{obst},jo2} \tag{40b}$$

$$x^p_{k+j|k} \ge x_{\text{high},o} + \alpha_j - M\, b^p_{\text{obst},jo3} \tag{40c}$$

$$y^p_{k+j|k} \ge y_{\text{high},o} + \alpha_j - M\, b^p_{\text{obst},jo4} \tag{40d}$$

$$\sum_{i=1}^{4} b^p_{\text{obst},joi} \le 3 \tag{40e}$$

where M is a large number to relax the constraints in (40a)–(40d), and o denotes the index of the obstacle. The logical constraint (40e) requires at least one constraint in (40a)–(40d) be active. Note that the parameter α_j tightens the constraints.

The output constraint (9) also includes the bound on speed and inputs. Let vectors $\boldsymbol{r}, \boldsymbol{v}$, and $\boldsymbol{a}$ respectively represent position, velocity, and acceleration input in the inertia frame. A set of n_d linear constraints approximates the two-norm bounded constraints on the acceleration and velocity vectors, which in turn limits the maximum turning rate

$$\begin{bmatrix}\cos\theta_m, \ \sin\theta_m\end{bmatrix} \boldsymbol{v}^p_{k+j|k} \leq v_{\text{max}} - \sqrt{2}\beta_j \tag{41a}$$

$$\begin{bmatrix}\cos\theta_m, \ \sin\theta_m\end{bmatrix} \boldsymbol{a}^p_{k+j|k} \leq a_{\text{max}} - \sqrt{2}\gamma_j \tag{41b}$$

$$\theta_m = \frac{2\pi m}{n_d}, \quad \forall m = 1, \dots, n_d$$

The scaling $\sqrt{2}$ is multiplied to account for the effect of the infinity-norm bounded disturbance on the two-norm bounded constraints. The minimum speed constraint is non-convex and requires n_v binary variables to express in MILP

$$\begin{bmatrix}\cos\theta_m, \ \sin\theta_m\end{bmatrix} \boldsymbol{v}^p_{k+j|k} \geq v_{\text{min}} + \sqrt{2}\beta_j - 2v_{\text{max}} b^p_{\text{vel},jm} \tag{42a}$$

$$\sum_{m=1}^{n_v} b^p_{\text{vel},jm} \leq n_v - 1 \tag{42b}$$

For vehicles $q\,(> p)$, the subscript j in the the constraint tightening parameters $\alpha_j, \beta_j, \gamma_j$ must be replaced with $j+1$, as shown in (28)–(29).

C. Invariance Constraints (12)

For the fixed-wing aircraft, the vehicle has an option to enter left or right loiter circle from the terminal states. The center of the left and right safety circles are

$$\boldsymbol{O}^p_{\text{L}} = \boldsymbol{r}^p_{k+N|k} + R\left(\frac{\pi}{2}\right) \frac{\rho}{v_{\text{max}} - \beta_N} \boldsymbol{v}^p_{k+N|k} \tag{43a}$$

$$\boldsymbol{O}^p_{\text{R}} = \boldsymbol{r}^p_{k+N|k} + R\left(-\frac{\pi}{2}\right) \frac{\rho}{v_{\text{max}} - \beta_N} \boldsymbol{v}^p_{k+N|k} \tag{43b}$$

where $R(\theta)$ is a rotation matrix of angle θ, and ρ is the radius of the turning circle given by

$$\rho = \frac{(v_{\text{max}} - \beta_N)^2}{a_{\text{max}} - \gamma_N} \times \frac{v_{\text{max}} - \beta_N}{v_{\text{min}} + \beta_N}.$$

The second term accounts for the variability of the terminal speed $\|\boldsymbol{v}^p_{k+N|k}\|$. The binary variable b^p_{left} chooses either the left or right safety circle

$$\begin{aligned} &\boldsymbol{O}^p_{\text{L}} - 2(\rho + \alpha_{N-1})(1 - b^p_{\text{left}}) \\ &\leq \boldsymbol{O}^p \leq \boldsymbol{O}^p_{\text{L}} + 2(\rho + \alpha_{N-1})(1 - b^p_{\text{left}}) \end{aligned} \tag{44a}$$

$$\begin{aligned} &\boldsymbol{O}^p_{\text{R}} - 2(\rho + \alpha_{N-1}) b^p_{\text{left}} \\ &\leq \boldsymbol{O}^p \leq \boldsymbol{O}^p_{\text{R}} + 2(\rho + \alpha_{N-1}) b^p_{\text{left}} \end{aligned} \tag{44b}$$

where the cost-to-go (18) is evaluated from $\boldsymbol{O}^p$. With the notation $\boldsymbol{O}^p = [x^p_{\text{center}}, \ y^p_{\text{center}}]^{\text{T}}$, the obstacle avoidance constraints of the safety circle are written as

$$\forall o: \quad x^p_{\text{center}} \leq x_{\text{low},o} - (\rho + \alpha_{N-1}) + M\, b^p_{\text{circ-obst},o1} \tag{45a}$$

$$y^p_{\text{center}} \leq y_{\text{low},o} - (\rho + \alpha_{N-1}) + M\, b^p_{\text{circ-obst},o2} \tag{45b}$$

$$x^p_{\text{center}} \geq x_{\text{high},o} + (\rho + \alpha_{N-1}) - M\, b^p_{\text{circ-obst},o3} \tag{45c}$$

$$y^p_{\text{center}} \geq y_{\text{high},o} + (\rho + \alpha_{N-1}) - M\, b^p_{\text{circ-obst},o4} \tag{45d}$$

$$\sum_{i=1}^{4} b^p_{\text{circ-obst},oi} \leq 3 \tag{45e}$$

D. *Interconnected Constraints* (11)

Over the planning horizon, the coupling constraints include vehicle avoidance constraints

$$x^p_{k+j|k} \le x^q_{k+j|k} - d^{pq}_{\text{total}} + M\, b^{pq}_{\text{veh},j1} \tag{46a}$$

$$y^p_{k+j|k} \le y^q_{k+j|k} - d^{pq}_{\text{total}} + M\, b^{pq}_{\text{veh},j2} \tag{46b}$$

$$x^p_{k+j|k} \ge x^q_{k+j|k} + d^{pq}_{\text{total}} - M\, b^{pq}_{\text{veh},j3} \tag{46c}$$

$$y^p_{k+j|k} \ge y^q_{k+j|k} + d^{pq}_{\text{total}} - M\, b^{pq}_{\text{veh},j4} \tag{46d}$$

$$\sum_{i=1}^{4} b^{pq}_{\text{veh},ji} \le 3 \tag{46e}$$

where

$$d^{pq}_{\text{total}} = \begin{cases} 2d + 2\alpha_j, & q < p \\ 2d + \alpha_j + \alpha_{j+1}, & q > p \end{cases}$$

where d is the size of the vehicle region that other vehicle must not enter. Beyond the planning horizon, constraints on the safety circles ensure the vehicle avoidance

$$x^p_{\text{center}} \le x^q_{\text{center}} - 2(\rho + d + \alpha_{N-1}) + M\, b^{pq}_{\text{circ},1} \tag{47a}$$

$$y^p_{\text{center}} \le y^q_{\text{center}} - 2(\rho + d + \alpha_{N-1}) + M\, b^{pq}_{\text{circ},2} \tag{47b}$$

$$x^p_{\text{center}} \ge x^q_{\text{center}} + 2(\rho + d + \alpha_{N-1}) - M\, b^{pq}_{\text{circ},3} \tag{47c}$$

$$y^p_{\text{center}} \ge y^q_{\text{center}} + 2(\rho + d + \alpha_{N-1}) - M\, b^{pq}_{\text{circ},4} \tag{47d}$$

$$\sum_{i=1}^{4} b^{pq}_{\text{circ},i} \le 3 \tag{47e}$$

E. *Objective Function*

The objective function (18) uses a binary variable $\boldsymbol{b}_{\text{vis}}$ to select one visible point $\boldsymbol{r}_{\text{vis}}$ from a list of cost points, from which the cost-to-go is known. Let $\boldsymbol{r}_{\text{cp},i}$ denote the i^{th} cost point and $i = 1, \ldots, n_{\text{cp}}$ where n_{cp} is a number of cost points stored in the cost map. Then,

$$\boldsymbol{r}^p_{\text{vis}} = \sum_{i=1}^{n_{\text{cp}}} b^p_{\text{vis},i}\ \boldsymbol{r}^p_{\text{cp},i} \tag{48a}$$

$$\sum_{i=1}^{n_{\text{cp}}} b^p_{\text{vis},i} = 1 \tag{48b}$$

$$\tilde{f}^p(\boldsymbol{r}^p_{\text{vis}}) = \sum_{i=1}^{n_{\text{cp}}} b^p_{\text{vis},i}\ \tilde{f}^p(\boldsymbol{r}^p_{\text{cp},i}) \tag{48c}$$

$$J^p \ge [\cos\theta_m, \sin\theta_m](\boldsymbol{O}^p - \boldsymbol{r}^p_{\text{vis}}) + \tilde{f}^p(\boldsymbol{r}^p_{\text{vis}}), \quad \forall m \tag{48d}$$

where the cost $\tilde{f}^p(\boldsymbol{r}^p_{\text{cp},i})$ from each cost point to the target of vehicle p is calculated prior to MILP and is constant in MILP.

To ensure the selected cost point $\boldsymbol{r}^p_{\text{vis}}$ is visible from a point $\boldsymbol{O}^p$ in the invariant set Q^p_k, obstacle avoidance constraints are enforced on n_{int} interpolation points that are placed on the line connecting $\boldsymbol{r}^p_{\text{vis}}$ and $\boldsymbol{O}^p$

$$\mu_l x^p_{\text{center}} + (1 - \mu_l) x^p_{\text{vis}} \le x_{\text{low},o} - \alpha_{N-1} + M\, b^p_{\text{int},lo1} \tag{49a}$$

$$\mu_l y^p_{\text{center}} + (1 - \mu_l) y^p_{\text{vis}} \le y_{\text{low},o} - \alpha_{N-1} + M\, b^p_{\text{int},lo2} \tag{49b}$$

$$\mu_l x^p_{\text{center}} + (1 - \mu_l) x^p_{\text{vis}} \ge x_{\text{high},o} + \alpha_{N-1} - M\, b^p_{\text{int},lo3} \tag{49c}$$

$$\mu_l y^p_{\text{center}} + (1 - \mu_l) y^p_{\text{vis}} \ge y_{\text{high},o} + \alpha_{N-1} - M\, b^p_{\text{int},lo4} \tag{49d}$$

$$\sum_{i=1}^{4} b^{p}_{\text{int},loi} \le 3 \tag{49e}$$

$$\mu_l = \frac{l}{n_{\text{int}}}, \qquad l = 1,\dots,n_{\text{int}}.$$

F. Decision Variables

The inputs are constrained to be

$$\forall q: \quad \left[\boldsymbol{u}^{q\;T}_{k|k}, \cdots, \boldsymbol{u}^{q\;\;T}_{k+N-1|k} \right]^T = \bar{U}^q_k + T^q \xi^q. \tag{50}$$

The binary variables are fixed if the superscript does not include p, except for b^q_{left} which selects the left/right safety circle

$$\forall q: \qquad b^q_{\text{obst}} = \bar{b}^q_{\text{obst}} \tag{51a}$$

$$b^q_{\text{circ-obst}} = \bar{b}^q_{\text{circ-obst}} \tag{51b}$$

$$b^q_{\text{vis}} = \bar{b}^q_{\text{vis}} \tag{51c}$$

$$b^q_{\text{int}} = \bar{b}^q_{\text{int}} \tag{51d}$$

$$b^{qr}_{\text{veh}} = \bar{b}^{qr}_{\text{veh}} \quad \forall r \neq p,\; r \neq q \tag{51e}$$

$$b^{qr}_{\text{circ-circ}} = \bar{b}^{qr}_{\text{circ-circ}}, \quad \forall r \neq p,\; r \neq q \tag{51f}$$

In summary, the MILP implementation of subproblem P^p_k is to minimize (6) subject to (7)–(8), (40)–(51).

References

[1] T. McLain, P. Chandler, S. Rasmussen, and M. Pachter, "Cooperative Control of UAV Rendezvous," in *Proceedings of the IEEE American Control Conference*, Arlington, VA, June 2001, pp. 2309 – 2314.

[2] R. Saber, W. Dunbar, and R. Murray, "Cooperative control of multi-vehicle systems usiing cost graphs and optimization," in *Proceedings of the IEEE American Control Conference*, 2003.

[3] S. Butenko, R. Murphey, and P. P. (Eds.), *Recent Developments in Cooperative Control and Optimization*. Kluwer Academic Publishers, 2004, vol. 3.

[4] D. Grundel, R. Murphey, and P. P. (Eds.), *Theory and Algorithms for Cooperative Systems*, ser. Series on Computers and Operations Research. World Scientific Publishing Co., 2004, vol. 4.

[5] O. the Secretary of Defense, "Unmanned systems roadmap 2007–2032," Tech. Rep., December 2007. [Online]. Available: http://www.acq.osd.mil/usd/Unmanned Systems Roadmap.2007-2032.pdf

[6] M. G. Singh and A. Titli, *Systems: Decomposition, Optimisation and Control*. Pergamon Press, 1978.

[7] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, vol. 2, pp. 44–52, 2002.

[8] D. Jia and B. Krogh, "Min-max feedback model predictive control for distributed control with communication," in *Proceedings of the IEEE American Control Conference*, 2002, pp. 4507–45.

[9] W. B. Dunbar and R. Murray, "Model Predictive Control of Coordinated Multi-vehicle Formations," in *Proceedings of the IEEE Conference on Decision and Control*, 2002, pp. 4631–4636.

[10] D. Shim, H. J. Kim, and S. Sastry, "Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots," in *Proceedings of the IEEE Conference on Decision and Control*, 2003, pp. 3621–3626.

[11] S. L. Waslander, G. Inalhan, and C. J. Tomlin, "Decentralized optimization via Nash bargaining," in *Fourth Annual Conference on Cooperative Control and Optimization*, D. Grundel, R. Murphy, and P. M. Pardalos, Eds., 2003, pp. 565–583.

[12] T. Keviczky, F. Borrelli, and G. Balas, "A study on decentralized receding horizon control for decoupled systems," in *Proceedings of the IEEE American Control Conference*, 2004, pp. 4921–4926.

[13] T. Schouwenaars, J. How, and E. Feron, "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, August 2004, pp. 6820–6825.

[14] M. Flint, M. Polycarpou, and E. Fernandez-Gaucherand, "Cooperative path-planning for autonomous vehicles using dynamics programming," in *15th IFAC World Congress*, 2002, pp. 1694–1699.

[15] D. P. Bertsekas, "Separable Dynamic Programming and Approximate Decomposition Methods," Laboratory for Information and Decision Systems, MIT, Report 2684, Feb 2006.

[16] A. N. Venkat and J. B. Rawlings and S. J. Wright, "Plant-wide optimal control with decentrlized MPC," Department of Chemical and Biological Engineering, University of Wisconsin, Tech. Rep., 2004.

[17] G. Inalhan, D. M. Stipanovic, and C. J. Tomlin, "Decentralized Optimization, with Application to Multiple Aircraft Coordination," in *Proceedings of the IEEE Conference on Decision and Control*, December 2002.

[18] A. Richards and J. How, "A Decentralized Algorithm for Robust Constrained Model Predictive Control," in *Proceedings of the IEEE American Control Conference*. Boston, MA: IEEE, 2004, pp. 4261–4266.

[19] W. Dunbar and R. Murray, "Receding horizon control of multi-vehicle formation: A distributed implementation," in *Proceedings of the IEEE Conference on Decision and Control*, 2004.

[20] Y. Kuwata, A. Richards, T. Schouwenaars, and J. How, "Distributed Robust Receding Horizon Control for Multi-vehicle Guidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 627–641, 2007.

[21] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 19–33, January 2008.

[22] U. Ozguner and W. R. Perkins, "Optimal control of multilevel large-scale systems," *International Journal of Control*, vol. 28, no. 6, pp. 967–980, 1978.

[23] P. Heiskanen, "Decentralized method for computing Pareto solutions in multiparty negotiations," *European journal of Operational Research*, vol. 117, pp. 578–590, 1999.

[24] A. Venkat, J. Rawlings, and S. Wright, "Stability and optimality of distributed model predictive control," in *IEEE Conference on Decision and Control, and the European Control Conference*, 2005.

[25] A. Richards and J. How, "Decentralized Model Predictive Control of Cooperating UAVs," in *Proceedings of the IEEE Conference on Decision and Control*, December 2004, pp. 4286–4291.

[26] Y. Kuwata, A. Richards, and J. How, "Robust Receding Horizon Control using Generalized Constraint Tightening," in *Proceedings of the IEEE American Control Conference*, New York, NY, July 2007, pp. 4482–4487.

[27] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Mathematical Problems in Engineering: Theory, Methods and Applications*, vol. 4, pp. 317–367, 1998.

[28] T. Schouwenaars, J. How, and E. Feron, "Receding Horizon Path Planning with Implicit Safety Guarantees," in *Proceedings of the IEEE American Control Conference*. Boston, MA: IEEE, July 2004, pp. 5576–5581.

[29] J. Bellingham, A. Richards, and J. How, "Receding Horizon Control of Autonomous Aerial Vehicles," in *Proceedings of the IEEE American Control Conference*, Anchorage, AK, May 2002, pp. 3741–3746.

[30] A. Richards and J. P. How, "Robust Distributed Model Predictive Control," *International Journal of Control*, vol. 80, no. 9, pp. 1517–1531, September 2007.

[31] A. Richards, "Robust Constrained Model Predictive Control," Ph.D. dissertation, MIT, Department of Aeronautics and Astronautics, November 2004.

[32] M. Valenti, B. Bethke, G. Fiore, and J. How, "Indoor Multi-Vehicle Flight Testbed for Fault Detection, Isolation, and Recovery," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, Aug 2006.

[33] "UAV SWARM Project Website," December 2006. [Online]. Available: http://vertol.mit.edu/index.html

[34] "Vicon MX Systems," July 2006. [Online]. Available: http://www.vicon.com/products/viconmx.html

[35] Draganfly Innovations Inc., "Draganfly V Ti Pro website," January 2006. [Online]. Available: http://www.rctoys.com/draganflyer5tipro.php

[36] A. Richards, "Robust Model Predictive Control for Time-Varying Systems," in *Proceedings of the IEEE Conference on Decision and Control*, 2005.